ON THE CONVEXITY OF RIGHT-CLOSED SETS AND ITS APPLICATION TO LIVENESS ENFORCEMENT IN PETRI NETS

BY

EHSAN SALIMI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Associate Professor Ramavarapu Sreenivas, Chair, Director of Research
Professor Tamer Basar
Associate Professor Carolyn L. Beck
Associate Professor Negar Kiyavash
Professor Rakesh Nagi

Abstract

A set of n-dimensional integral vectors, $\Omega \subseteq \mathcal{N}^n$, is said to be *right-closed* if for any $\mathbf{x} \in \Omega$, any vector $\mathbf{y} \geq \mathbf{x}$ also belongs to it. An integral-set $\Omega \subseteq \mathcal{N}^n$ is convex if and only if there is a convex set $\mathbf{C} \subseteq \mathcal{R}^n$ such that $\Omega = Int(\mathbf{C})$, where $Int(\bullet)$ denotes the integral points in the set argument. In this dissertation, we show that the problem of verifying convexity of a right-closed set is decidable. Following this, we present a polynomial time, LP-based algorithm, for verifying the convexity of a right-closed set of integral vectors, when the dimension n is fixed. This result is to be viewed against the backdrop of the fact that checking the convexity of a real-valued, geometric set can only be accomplished in an approximate sense; and, the fact that most algorithms involving sets of real-valued vectors do not apply directly to their integral counterparts. Also, we discuss a grid-search based algorithm for verifying the convexity of such a set, although not a polynomial time procedure, it is a method that verifies the convexity of right-closed sets in a reasonable time complexity.

On the application side, right-closed sets feature in the synthesis of *Liveness Enforcing Supervisory Policies* (LESPs) for a large family of *Petri Nets* (PNs). For any PN structure N from this family, the set of *initial markings*, $\Delta(N)$, for which there is a LESP, is right-closed. A LESP determines the *transitions* of a PN that are to be permitted to fire at any marking in such a manner that, irrespective of the past, every transition can be fired at some marking in the future. A system that is modeled by a live PN does not experience livelocks, which serves as the motivation for investigating implementation paradigms for LESPs in practice.

If a transition is prevented from firing at a marking by a LESP, and *all* LESPs, irrespective of the implementation-paradigm that is chosen, prescribe the same control for the marking, then it is a *minimally restrictive* LESP. It is possible to synthesize the minimally restrictive LESP for any instance

N of the aforementioned family that uses the right-closed set of markings $\Delta(N)$. The literature also contains an implementation paradigm called *invariant-based monitors* for liveness enforcement in PNs. This paradigm is popular due to the fact that the resulting supervisor can be directly incorporated into the semantics of the PN model of the controlled system. In this work, we show that there is an invariant-based monitor that is equivalent to the minimally restrictive LESP that uses the right-closed set $\Delta(N)$ if and only if $\Delta(N)$ is convex. This result serves as the motivation behind exploring the convexity of right-closed sets.

To Mom, Dad, Amir and Neda for their endless love and support.

Acknowledgment

I would like to express my deepest gratitude to my advisor, Dr. Ramavarapu Sreenivas, whom I owe most of what I have learned through the course of my doctoral study at the University of Illinois. His constant encouragement and understanding, as well as our many long discussions and his helpful comments, have been the source of inspiration for me. Being his student and working with him, has been an honor for me and will remain this way.

I also want to extend my appreciation to my committee members: Dr. Tamer Basar, Dr. Rakesh Nagi, Dr. Carolyn L.Beck and Dr. Negar Kiyavash, for agreeing to serve on my committee. I truly appreciate our fruitful discussions regarding my work and possible future research directions. I also want to thank them all for their helpful comments and suggestions for preparing this manuscript.

I would like to give special thanks to my parents for their endless supports during all these years, from miles away. It would have not been possible for me to graduate without their encouragement. I also would like to thank my brother, Amir, who was the engine behind my motivation for successfully finishing my PhD and always was on my side during tough times.

At the end, I would like to thank my friends for all the care and support they provided. While the limits of space will not permit me to thank them all individually, I hope they will find some bit of themselves expressed in the pages that follow.

Table of Contents

Chapter 1

Introduction

A *right-closed* (or *upward-closed*) set [1], $\Omega \subseteq \mathcal{N}^n$, of n-dimensional integral-vectors satisfies the condition

$$(\mathbf{x} \in \Omega) \wedge (\mathbf{y} \geq \mathbf{x}) \Rightarrow \mathbf{y} \in \Omega.$$

Every right-closed set of integral vectors, $\Omega \in \mathcal{N}^n$, contains a finite set called *minimal elements*, $min(\Omega) \subset \Omega$, such that:

1. $\forall \mathbf{x} \in \Omega, \exists \mathbf{y} \in \min(\Omega)$ such that $\mathbf{x} \geq \mathbf{y}$, and

2. if $\exists \mathbf{x} \in \Omega, \exists \mathbf{y} \in \min(\Omega)$ such that $\mathbf{y} \geq \mathbf{x}$, then $\mathbf{x} = \mathbf{y}$.

Any integer-valued right-closed set can be constructed uniquely based on its minimal elements. If $\Omega_1, \Omega_2 \subseteq \mathcal{N}^n$ are two right-closed sets of n-dimensional integral vectors, then $(\Omega_1 = \Omega_2) \Leftrightarrow (\min(\Omega_1) = \min(\Omega_2))$.

What motivates us to take a deeper look at the geometric properties of these sets, convexity in particular, is that the set of *initial markings* of a large class of *Petri Nets* (PNs), for which there exists a *liveness enforcing supervisory policy* (LESP) is right-closed. This right-closed set plays a critical role in the synthesis and implementation of LESPs. By studying the geometric properties of right-closed sets, we hope to develop a better understanding of synthesis procedures for LESPs for PN models.

Our discourse centers on a class of dynamical systems called *Discrete-Event/Discrete-State* (DEDS) systems. The (discrete-)states of these systems have a logical, as opposed to numerical, interpretation. At each state, there are potential (discrete-)events that can occur, the occurrence of any one of them would change the state of the system, which then results in a new set of potential events, and this process can be repeated as often as necessary. DEDS systems are regulated by a *supervisory policy*, which determines which event is to be permitted at each state, in such a manner that some behavioral specification is satisfied. Our focus is on the synthesis and implementation of *liveness enforcing supervisory policies* (LESPs). A DEDS system is *live* [2], if irrespective of the past, every event can be executed, not necessarily immediately, in the future. A live DEDS system does not experience *livelocks*, and serves as the main motivation for investigations into the synthesis and implementation of LESPs. A DEDS system is in a *livelocked*-state, if *some* event has entered into a state of suspended-animation for perpetuity. If *every* event of the DEDS system is in a state of suspended animation, the DEDS system is *deadlocked*. A livelock-free DEDS system does not have deadlocked-states, but a deadlock-free DEDS system can still experience livelocks. Livelock freedom is harder to achieve, compared to deadlock freedom.

There are two paradigms for the supervisory control of DEDS systems: (1) *State-Based* supervisory control, and (2) *Event-Based* supervisory control. In event-based control, the supervisory policy uses the string of past event-occurrences to decide the appropriate control action for any given instant. References [3–12] deal with various aspects of event-based supervisory control of DEDS systems. In contrast, the control action of a state-based supervisory policy is determined by the current discrete-state of the DEDS system (cf. [13, 14]). These two paradigms are equivalent, but based on specific applications, one paradigm might be preferred over the other.

DEDS systems with asynchronous events are usually modeled by PNs [15–17]. Each task is represented by a *transition* and the state of the system is represented by a *marking*. These terms are formally defined in the next chapter. Examples of such systems can be manufacturing systems, computer operating systems, and command, control, communications, computers and intelligence (C^4I) systems. We consider state-based LESPs for PN models of DEDS systems in this thesis.

As an arbitrary PN model is not necessarily *live*, it is important to design a *supervisory policy* [3], that makes sure that the system will not be trapped in a *livelock*-state. In the context of computer operating systems, a *livelock*-state would occur once some task hangs, while the rest of the tasks proceed without impediment. In many cases, rebooting the system is the only way to resolve this problem. Such an action can be costly when a safety critical system like avionics is involved, which motivates us to take a deeper look at this particular problem. We aim to design a LESP which makes sure that none of the processes enter to a perpetual state of livelock along with two additional conditions:

1. Tasks external to the systems, which are modeled as *uncontrollable transitions* in the PN, should not be prevented from execution, and

2. The LESP should be *minimally restrictive*, meaning if the LESP prevents a (controllable) transition from execution, all the other LESPs should prevent the same transition from execution as well.

The LESP is basically a set of rules which determines which *controllable transition* is not allowed to be executed at a particular marking.

The LESP can be implemented by altering the original PN and augmenting it with additional places called *monitors*. There are arcs between the transitions in the PN and the monitors. The monitor is initialized appropriately, after which the augmented PN functions as if it were a PN. That is, the LESP's functionality is absorbed into the semantics of the PN. This is the main reason behind the popularity of monitor-based control for PNs. There is a specific technique, called *invariant-based* monitors that has been extensively studied in the DEDS literature. The disadvantage of this method is that applying the invariant-based monitors often times will not result in a minimally restrictive policy, meaning it will stop events from occurring unnecessarily.

To tackle this problem we are going to focus on a family of PNs that is defined in Chapter 3. This family is large and includes a class of PNs called *Free Choice Petri Nets* (FCPNs), which are expressive enough to model the flow of material in manufacturing/logistic systems, and the flow of control in data-flow computation [18, 19]. This family of PNs possesses several desirable properties:

1. If there exists an LESP for a member of this family of PNs for a given initial marking, there is an LESP when the same PN is initialized with a larger initial marking. That is, the PN structure N belongs to the above mentioned family, and if $\Delta(N)$ denotes the set of initial markings of the PN structure N for which there is an LESP, then the set $\Delta(N)$ is right-closed.

2. As a consequence of the above property, the existence of a LESP for an arbitrary PN from this family is *decidable*. To appreciate the import of this property, it is worth pointing out that neither the existence, nor the non-existence of a LESP for arbitrary PNs is not even semi-decidable [20].

3. For any reachable marking in $\Delta(N)$, the minimally restrictive LESP prevents the firing of a controllable transition if the new marking that would result from its firing is not in $\Delta(N)$. An additional property of $\Delta(N)$ is that the firing of an uncontrollable transition at any marking in $\Delta(N)$ will always result in a new marking that is in $\Delta(N)$, that is the set $\Delta(N)$ is *control-invariant* [3] with respect to N .

In this dissertation, we first show that there is an invariant-based monitor that is equivalent to the minimally restrictive LESP identified above if and only if $\Delta(N)$ is convex. This result identifies the condition when the invariant-based monitor that enforces liveness is minimally restrictive. This motivates us to find an efficient algorithm which can test the convexity for a given right-closed set. It is worth mentioning that even testing the convexity for an arbitrary, real-valued, scalar-set can only be done in an approximate sense. For example, if a single point is deleted from a convex, real-valued, scalar-set, its convexity testing can take "forever" [21]. It gets worse for integral sets, as most of the convexity-testing algorithms on real-valued sets cannot be extended to integral sets. First, we show that testing the convexity of an arbitrary right-closed set is decidable, and there exists a procedure that can test the convexity of an arbitrary right-closed integral set. Following this we present an algorithm that can execute this test in in polynomial time, when the dimension of the integral vectors is fixed. The work that has been accomplished in this dissertation includes:

1. A characterization of the largest family of PN structures with a right-closed $\Delta(N)$-set for any member N of the family [22, 23];

2. A characterization of supervisory policies that enforces liveness in PNs that are "similar" [24];

3. A necessary and sufficient condition for the existence of a minimally restrictive invariant-based monitor that enforces liveness [25] for an arbitrary PN;

4. A procedure that shows that testing the convexity of a right-closed set of integral vectors is *decidable*[1]; and

5. A polynomial time algorithm for testing the convexity of a right-closed integral set.

These results make no assumptions regarding the *boundedness* of the PN models. Following reference [26], the results listed above can be serve as critical milestones in the synthesis of asymptotically efficient LESP synthesis procedures.

1.1 Outline of the thesis

Chapter 2 formally introduces Petri Nets (PNs), their properties, and the paradigm of supervisory control of PNs. Preliminaries of polyhedral theory and convexity theory is also presented in this chapter, along with a brief review of the procedure of counting the number of lattice points within a real-valued polytope. This enumeration procedure is used to analyze a grid-search procedure for convexity-testing in a subsequent chapter. In Chapter 3, we review the results on synthesizing liveness enforcing supervisory policy (LESP). We show that if the set of initial markings of a PN structure N for which there is a LESP, $\Delta(N)$, is convex, then we can use a procedure known to the literature as invariant-based-monitors to enforce liveness in a class of PNs with where the $\Delta(N)$-set is right-closed. This is the motivation behind the testing of convexity of right-closed sets. In Chapter 4, we show that testing the convexity for an arbitrary right-closed set is decidable. Chapter 5, introduces two algorithms for testing convexity of right-closed integral sets. The first algorithm tests convexity in polynomial time and second algorithm, although not polynomial, still can perform the same test in an efficient manner. Chapter 6 discusses possible heuristic methods to tackle the problem along with illustrative examples. Chapter 7 suggests few possible directions to be explored in the future and Chapter 8 concludes this dissertation.

[1] That is, there is a program P that takes $\min(\Omega)$ as input, for an arbitrary right-closed set $\Omega \subseteq \mathcal{N}^n$, and eventually halts after producing an output of unity (resp. zero) if Ω is convex (resp. not convex).

Chapter 2

Notation and Definitions

In this chapter, some basic concepts and definitions involving *Petri Nets* (PNs) are reviewed. This is followed by a brief review of concepts from polyhedral theory. Specific attention is paid to the problem of counting the number of lattice points in real-valued polyhedra, which is used in the analysis of a grid-search algorithm for convexity testing in Chapter 5.

2.1 Petri Net

A *Petri Net Structure* is a weighted, directed bipartite graph with a two classes of vertices called *transitions* and *places*. It can be described as an ordered 4-tuple $N = (\Pi, T, \Phi, \Gamma)$ where $\Pi = \{p_1, p_2, \cdots, p_n\}$ is the set of places, $T = \{t_1, t_2, \cdots, t_m\}$ describes the m possible transitions in the system, Φ is the collection of *arcs* where $\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$ and Γ is the weight function associated with each arc described as $\Gamma : \Phi \rightarrow \mathcal{N}$ where $\mathcal{N}$ is the set of non-negative integers. The amount of available resource at each place will be shown as black dots called *tokens*. The initial state of the system which is called *initial marking* is the number of tokens in each place at the beginning of the process, which is defined by *initial marking function* as $\mathbf{m}^0 : \Pi \rightarrow \mathcal{N}$. The state of the system at each time will be defined similar to the initial marking function as $\mathbf{m} : \Pi \rightarrow \mathcal{N}$. The PN is called *ordinary* if the weight of each arc is 1, otherwise it is *general*. Arcs with weight of 1 will not be labeled for simplicity. We use the term *Petri Net* (PN) to denote a Petri net structure together with its initial marking: $N(\mathbf{m}^0)$.

The notations ${}^\bullet x$ and $x^\bullet$ will be used to denote sets $\{y \mid (y, x) \in \Phi\}$ and $\{y \mid (x, y) \in \Phi\}$ respectively. A transition (event), $t \in T$ in order to *fire* (occur) at a marking $\mathbf{m}^i$ should be *enabled*, meaning $\forall p \in {}^\bullet t$, $\mathbf{m}^i(p) \geq \Gamma(p, t)$. The set of enabled transitions at marking $\mathbf{m}^i$ will be denoted by $T_e(N, \mathbf{m}^i)$.

The marking of the system after firing the enabled transition t will change to $\mathbf{m}^{i+1}(p) = \mathbf{m}^i(p) - \Gamma(p,t) + \Gamma(t,p)$. If the firing of a string of transitions $\sigma \in T^*$, starting from a marking $\mathbf{m}^i$ results in a marking $\mathbf{m}^j$, we denote it as $\mathbf{m}^i \xrightarrow{\sigma} \mathbf{m}^j$. The set of markings *reachable* for a PN with initial marking $\mathbf{m}^0$ will be denoted by $\Re(N, \mathbf{m}^0)$, which is the set of markings generated through a sequence of enabled transitions. The change in markings can also be described in a matrix form as each marking is understood as a non-negative integer vector. In those contexts, it is useful to define the *input matrix* **IN** and *output matrix*, **OUT** as two $n \times m$ matrices, where

$$\mathbf{IN}_{i,j} = \begin{cases} \Gamma(p_i, t_j) & \text{if } p_i \in^\bullet t_j \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{OUT}_{i,j} = \begin{cases} \Gamma(t_j, p_i) & \text{if } p_i \in t_j^\bullet \\ 0 & \text{otherwise} \end{cases}$$

The *incidence matrix* **C** then can be defined as: $\mathbf{C} = \mathbf{OUT} - \mathbf{IN}$. If we define $\sigma \in T^*$ as the string of transitions and $\mathbf{x}(\sigma)$ denotes the *Parikh vector* of σ, where the i-th element of the vector corresponds to the number of occurrences of transition t_i in the sequence, then the marking $\mathbf{m}^j$ which results from the firing of σ in the initial marking of $\mathbf{m}^i$ can be written as $\mathbf{m}^j = \mathbf{m}^i + \mathbf{C}\mathbf{x}(\sigma)$. A marking $\mathbf{m}^i$ is called *potentially reachable* from $\mathbf{m}^0$ if $\exists \mathbf{y} \in \mathcal{N}^m$ such that the equation $\mathbf{C}\mathbf{y} = (\mathbf{m}^i - \mathbf{m}^0)$ is satisfied. While every reachable marking is also potentially reachable, there can be potentially reachable markings that cannot be reachable.

As an illustration, consider the PN $N_1(\mathbf{m}_1^0) = (\Pi_1, T_1, \Phi_1, \Gamma_1)$ shown in figure 2.1(a). The place-set $\Pi_1 = \{p_1, p_2\}$, the transition-set $T_1 = \{t_1, t_2, t_3\}$, and the set of arcs $\Phi_1 = \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_1), (p_1, t_3), (p_2, t_3)\}$. Each arc of this PN has a unitary weight associated with it. The initial marking $\mathbf{m}_1^0$ can be interpreted as a vector $(1\ 0)^T$; or, as a function where $\mathbf{m}_1^0(p_1) = 1$ and $\mathbf{m}_1^0(p_2) = 0$. From the definition, it follows that $T_e(N_1, \mathbf{m}_1^0) = \{t_1\}$, and the firing of t_1 at marking $\mathbf{m}_1^0$ will result in a new marking $\mathbf{m}_1^1 = (0\ 1)^T$. The $\mathbf{IN}, \mathbf{OUT}$ and $\mathbf{C}$ matrices for this PN are:

$$\mathbf{IN} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \mathbf{OUT} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \text{ and } \mathbf{C} = \begin{pmatrix} -1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix}.$$

If we interpret markings as vectors, the process of firing $t_1 \in T_e(N_1, \mathbf{m}_1^0)$ can be represented by the equation:

$$\underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{m}_1^1} = \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\mathbf{m}_1^0} + \underbrace{\begin{pmatrix} -1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix}}_{\mathbf{C}} \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\mathbf{x}(t_1)}$$

Similarly, the process of firing $t_2 \in T_e(N_1, \mathbf{m}_1^1)$ can be represented as

$$\underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\mathbf{m}_1^0} = \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{m}_1^1} + \underbrace{\begin{pmatrix} -1 & 1 & -1 \\ 1 & -1 & -1 \end{pmatrix}}_{\mathbf{C}} \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\mathbf{x}(t_2)}$$

The function- and vector-interpretation of the marking will be used interchangeably in this thesis. The context will indicate the appropriate interpretation.

It is not hard to see that the marking $(m\ n)^T$ for $m, n \geq 1$ is not potentially reachable for the PN $N_1(\mathbf{m}_1^0)$ as $\Re(N_1, \mathbf{m}_1^0) = \{(1\ 0)^T, (0\ 1)^T\}$; this would mean that the marking $(m\ n)^T, m, n \geq 1$ is not reachable from $\mathbf{m}_1^0$ either. Consequently, there can be no marking that is reachable from $\mathbf{m}_1^0$ where the transition t_3 can fire.

A transition $t \in T$ in a PN $N(\mathbf{m}^0)$ is said to be live, if

$$\forall \mathbf{m}^1 \in \Re(N, \mathbf{m}^0), \exists \mathbf{m}^2 \in \Re(N, \mathbf{m}^1), \text{ such that } t \in T_e(N, \mathbf{m}^2).$$

Transitions t_1 and t_2 are live in $N_1(\mathbf{m}_1^0)$ of figure 2.1(a), but transition t_3 is not. In contrast, none of the transitions in the PN $N_2(\mathbf{m}_2^0)$ shown in figure 2.1(b) are live, after t_1 is fired. This is because, $(0\ 1\ 0\ 1\ 0\ 0)^T \in \Re(N_2, \mathbf{m}_2^0)$ and $T_e(N_2, (0\ 1\ 0\ 1\ 0\ 0)^T) = \emptyset$. If all transitions in a PN are live, then we say the PN is live.

2.2 Supervisory Policy for PNs

Petri Nets are regulated by a *supervisory policy*, which determines which event is to be permitted at each state, in such a manner that some behavioral specification is satisfied. Some of the transitions represent local activities that can be prevented, while others that are external to system cannot be prevented by the *supervisor*. As an illustration, transitions that represent failures cannot be prevented from firing, but transitions that represent admission of an entity into a system, can be prevented when the system is near full-capacity. Consequently, the set of transitions are partitioned into the set of *uncontrollable* transitions T_u and, the set of *controllable* transitions T_c, where $T = T_u \cup T_c$ and $T_u \cap T_c = \emptyset$. Therefore, uncontrollable transitions cannot prevented from firing by the supervisor and in graphical presentation will be showed by unfilled boxes.

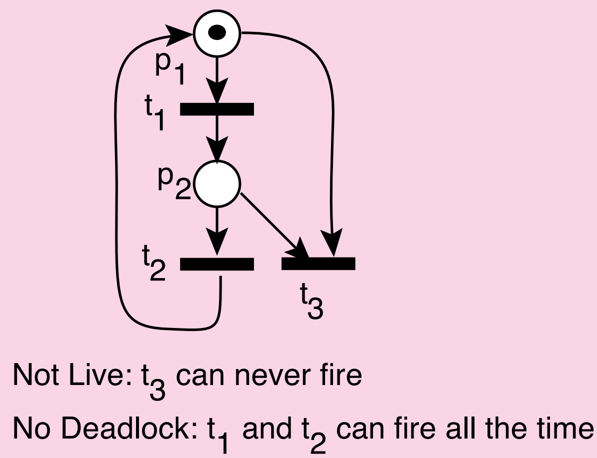
The *supervisory policy* can be described as a function $\mathcal{P} : \mathcal{N}^n \times T \to \{0, 1\}$, which returns 0 or 1 for each transition at each reachable marking. For each transition $t \in T$ at marking $\mathbf{m}^i$ if $t \in T_e(N, \mathbf{m}^i)$ then we say it is *state-enabled* and if $\mathcal{P}(\mathbf{m}^i, t) = 1$ we say it is *control-enabled.* For a transition to be allowed to fire under supervisory policy, it should be both state-enabled and control-enabled. It is important to mention that supervisory policy cannot prevent an uncontrollable transition from firing, meaning $\forall t \in T_u,\ \mathcal{P}(\mathbf{m}^i, t) = 1$.

A string of transitions $\sigma = t_1 t_2 \cdots t_k$, where $\forall j, t_j \in T$, starting at an initial marking $\mathbf{m}^i$, is called *valid firing string*, if two following conditions are met:

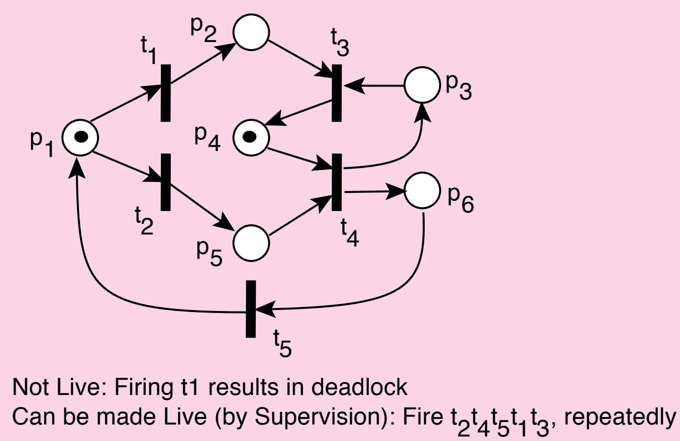
1. $t_1 \in T_e(N, \mathbf{m}^i), \mathcal{P}(\mathbf{m}^i, t_1) = 1$ and

2. for every transition t_j for $j \in \{1, 2, \cdots, k-1\}$ the firing of transition t_j produces a marking $\mathbf{m}^{i+j}$ where transition t_{j+1} is both state and control enabled.

The set of reachable markings for a net N from the initial marking $\mathbf{m}^0$ under the supervisory policy $\mathcal{P}$ will be denoted by $\Re(N, \mathbf{m}^0, \mathcal{P})$. It is also important to note that if a marking $\mathbf{m}^i$ is not potentially reachable



(a) Non-Live PN $N_1(\mathbf{m}_1^0)$



(b) Non-Live PN $N_2(\mathbf{m}_2^0)$

Figure 2.1: Figure (a) shows a PN $N_1(\mathbf{m}_1^0)$ which is not live, but it is not in deadlock position either. It cannot be made live by a LESP. Figure(b) shows a non-live PN $N_2(\mathbf{m}_2^0)$, where by a LESP, it can be turn to a live PN

from $\mathbf{m}^0$, then it's not reachable under the supervisory control as well, meaning $\mathbf{m}^i \notin \Re(N, \mathbf{m}^0, \mathcal{P})$.

Now we turn our attention to *liveness problem* [18, 27]. A PN $N(\mathbf{m}^0)$ is called *live* if

$$\forall t \in T, \forall \mathbf{m}^i \in \Re(N, \mathbf{m}^0), \exists \mathbf{m}^j \in \Re(N, \mathbf{m}^i) \text{ such that } t \in T_e(N, \mathbf{m}^j).$$

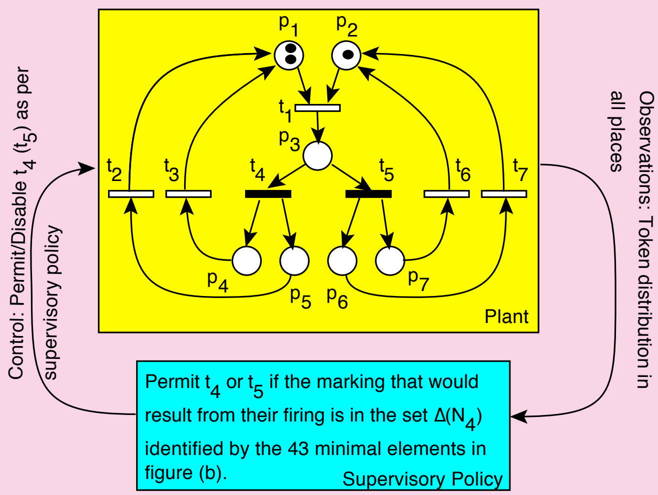
In other words, a Petri Net is live if no matter which marking has reached from the initial marking, all the transitions have the possibility to fire after executing a valid firing string of transitions. It is close to the concept of *livelock-free* operating systems. It is trivial to mention that we do not need a supervisory policy for a live PN. A PN that is not live, can be made live under the influence of a supervisory policy. A transition t_k is said to be *live under the policy* $\mathcal{P}$ if

$$\forall \mathbf{m}^i \in \Re(N, \mathbf{m}^0, \mathcal{P}), \exists \mathbf{m}^j \in \Re(N, \mathbf{m}^i, \mathcal{P}) \text{ such that } t_k \in T_e(N, \mathbf{m}^j) \text{ and } \mathcal{P}(\mathbf{m}^j, t_k) = 1.$$

If all transitions in $N(\mathbf{m}^0)$ are live under the policy $\mathcal{P}$, then we call $\mathcal{P}$ the *liveness enforcing supervisory policy* (LESP). If every LESP $\widehat{\mathcal{P}}$ for the net $N(\mathbf{m}^0)$ satisfies the condition $\mathcal{P}(\mathbf{m}^i, t) \geq \widehat{\mathcal{P}}(\mathbf{m}^i, t)$ for all $\mathbf{m}^i \in \mathbb{N}^n$ and $\forall t \in T$, then the policy $\mathcal{P}$ is said to be *minimally restrictive* policy [28, 29]. Figure 2.2(a) presents the minimally restrictive liveness enforcing supervisory policy for the plant PN $N_3(\mathbf{m}_3^0)$, which is not live in the absence of supervision. This policy permits the firing of controllable transitions t_4 and t_5 at any marking if and only if the new marking that might result from their firing is greater than or equal to one of the 43 vectors identified in figure 2.2(b). The structure of this LESP is explicated in the discussion that follows theorem 3.2.10 in subsequent text.

2.3 Liveness Enforcement using Monitors

Another popular method to enforce liveness in PNs is to augment the original PN structure N by adding extra places which are called *monitors*. Let $N(\mathbf{m}^0)$ be a PN, where $N = (\Pi, T, \Phi, \Gamma)$. The structure N can be augmented with the addition of extra places $\Pi_c = \{c_1, \ldots, c_k\}$ ($\Pi \cap \Pi_c = \emptyset$), or *monitors*, along with extra arcs $\Phi_c \subseteq (\Pi_c \times T) \cup (T \times \Pi_c)$ and their associated weights $\widehat{\Gamma} : \Phi_c \rightarrow \mathcal{N}^+$,



(a) Enforcing Liveness using supervisory policy

```
 1: ( 0 0 0 0 1 0 2 )
 2: ( 0 0 0 0 1 1 1 )
 3: ( 0 0 0 0 1 2 0 )
 4: ( 0 0 0 0 2 0 1 )
 5: ( 0 0 0 0 2 1 0 )
 6: ( 0 0 0 1 0 0 2 )
 7: ( 0 0 0 1 0 1 1 )
 8: ( 0 0 0 1 0 2 0 )
 9: ( 0 0 0 1 1 0 1 )
10: ( 0 0 0 1 1 1 0 )
11: ( 0 0 0 2 0 0 1 )
12: ( 0 0 0 2 0 1 0 )
13: ( 0 0 1 0 0 0 1 )
14: ( 0 0 1 0 0 1 0 )
15: ( 0 0 1 0 1 0 0 )
16: ( 0 0 1 1 0 0 0 )
17: ( 0 0 2 0 0 0 0 )
18: ( 0 1 0 0 1 0 1 )
19: ( 0 1 0 0 1 1 0 )
20: ( 0 1 0 0 2 0 0 )
21: ( 0 1 0 1 0 0 1 )
22: ( 0 1 0 1 0 1 0 )
23: ( 0 1 0 1 1 0 0 )
24: ( 0 1 0 2 0 0 0 )
25: ( 0 1 1 0 0 0 0 )
26: ( 0 2 0 0 1 0 0 )
27: ( 0 2 0 1 0 0 0 )
28: ( 1 0 0 0 0 0 2 )
29: ( 1 0 0 0 0 1 1 )
30: ( 1 0 0 0 0 2 0 )
31: ( 1 0 0 0 1 0 1 )
32: ( 1 0 0 0 1 1 0 )
33: ( 1 0 0 1 0 0 1 )
34: ( 1 0 0 1 0 1 0 )
35: ( 1 0 1 0 0 0 0 )
36: ( 1 1 0 0 0 0 1 )
37: ( 1 1 0 0 0 1 0 )
38: ( 1 1 0 0 1 0 0 )
39: ( 1 1 0 1 0 0 0 )
40: ( 1 2 0 0 0 0 0 )
41: ( 2 0 0 0 0 0 1 )
42: ( 2 0 0 0 0 1 0 )
43: ( 2 1 0 0 0 0 0 )
```

(b) 43 vectors

Figure 2.2: (a) A minimally restrictive Liveness Enforcing Supervisory Policy for the plant PN $N_3(\mathbf{m}_3^0)$, and (b) the 43 vectors used in the policy of figure (a) obtained using the software of reference [30].

to form a new structure $N_c = (\Pi \cup \Pi_c, T, \Phi \cup \Phi_c, \Gamma_c)$, where $\Gamma_c(\phi) = \Gamma(\phi)$ if $\phi \in \Phi$, and $\Gamma_c(\phi) = \widehat{\Gamma}(\phi))$ if $\phi \in \Phi_c$.

In subsequent text, when we deal with markings of N_c as $(n+k)$-dimensional vectors, we suppose the members of the place set of N_c are ordered as follows $\{p_1, \ldots, p_n, c_1, \ldots, c_k\}$, where $\Pi = \{p_1, \ldots, p_n\}$ and $\Pi_c = \{c_1, \ldots, c_k\}$. The initial token load of the monitors in Π_c are determined from the initial marking $\mathbf{m}^0$, according to $\Theta : \mathcal{N}^n \to \mathcal{N}^k$. The PN structure N_c with an initial marking of $((\mathbf{m}^0)^T \ \Theta(\mathbf{m}^0)^T)^T$ is represented as $N_c(\mathbf{m}^0, \Theta(\mathbf{m}^0))$. The set of markings that can be reached from the initial marking $((\mathbf{m}^0)^T \ \Theta(\mathbf{m}^0)^T)^T$ in N_c is denoted by $\Re(N_c, \mathbf{m}^0, \Theta(\mathbf{m}^0))$. Following the aforementioned convention, each $\mathbf{m} \in \Re(N_c, \mathbf{m}^0, \Theta(\mathbf{m}^0))$ can be interpreted as $\mathbf{m} = (\mathbf{m}_1^T \ \mathbf{m}_2^T)^T$, where the vector $\mathbf{m}_1 \in \mathcal{N}^n$ ($\mathbf{m}_2 \in \mathcal{N}^k$) corresponds to the token load of places in Π (Π_c). Since there might be arcs in Φ_c that originate from some $c_i \in \Pi_c$ to some uncontrollable transition $t_u \in T_u$, we must require $\forall \mathbf{m} \in \Re(N_c, \mathbf{m}^0, \Theta(\mathbf{m}^0))$,

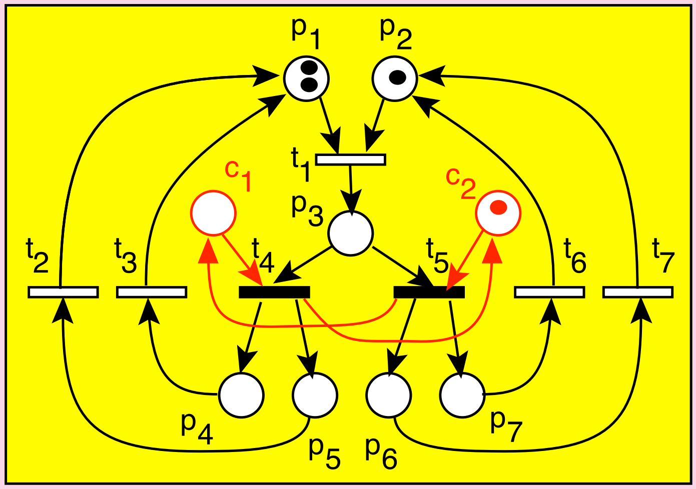
$$(\forall p \in ({}^\bullet t_u \cap \Pi),\ \mathbf{m}(p) \geq \Gamma_c((p, t_u))) \Rightarrow (\forall c \in ({}^\bullet t_u \cap \Pi_c), \mathbf{m}(c) \geq \Gamma_c((c, t_u)))\,.$$

That is, no uncontrollable transition is prevented from firing at some marking that is reachable in $N_c(\mathbf{m}^0, \Theta(\mathbf{m}^0))$ due to a lack of sufficient tokens in the monitors. The requirement, $(\Pi_c \times T_u) \cap \Phi_c = \emptyset$, is sufficient but not necessary, for the above condition to be true.

For $\mathbf{A} \in \mathcal{N}^{k \times n}$, $\mathbf{b} \in \mathcal{N}^k$, an initial marking $\mathbf{m}^0 \in \mathcal{N}^n$ where $\mathbf{A}\mathbf{m}^0 \geq \mathbf{b}$, and $\Theta(\mathbf{m}^0) = \mathbf{A}\mathbf{m}^0 - \mathbf{b}$, an *invariant-based* monitor ensures $\forall (\mathbf{m}_1^T \ \mathbf{m}_2^T)^T \in \Re(N_c, \mathbf{m}^0, \Theta(\mathbf{m}^0)), \mathbf{A}\mathbf{m}_1 \geq \mathbf{b}$ and $\mathbf{m}_2 = \mathbf{A}\mathbf{m}_1 - \mathbf{b} \geq \mathbf{0}$ [29]. That is, $\forall (\mathbf{m}_1^T \ \mathbf{m}_2^T)^T \in \Re(N_c, \mathbf{m}^0, \Theta(\mathbf{m}^0))$, the property $\mathbf{A}\mathbf{m}_1 \geq \mathbf{b}$, remains *invariant* for all reachable markings. When applicable, the invariant-based monitor is defined by the monitor incidence-matrix $\mathbf{AC}$, where $\mathbf{C}$ is the incidence matrix of the original PN structure N.

Liveness enforcement using invariant-based monitors seeks to augment the PN $N(\mathbf{m}^0)$ as described above, such that $N_c(\mathbf{m}^0, \Theta(\mathbf{m}^0))$ is live. When this objective is achieved, the influence of the monitors can be interpreted as an implicit definition of a LESP for the PN $N(\mathbf{m}^0)$. An illustrative example can found in figure 2.3. The PN $N_3(\mathbf{m}_3^0)$ shown in black in figure 2.3, which is the same PN that was shown in figure 2.2, is not live. However, when the structure of this PN is enhanced by the addition of monitors

$\{c_1, c_2\}$, along with arcs that involve the monitors (cf. $\{(c_1, t_4), (t_5, c_1), (c_2, t_5), (t_4, c_2)\}$ in figure 2.3), the resulting enhanced PN is live if a single token in placed in the monitor c_2.



(a) Monitor Placement

Figure 2.3: Liveness enforcement using monitor placement. The PN $N_3(\mathbf{m}_3^0)$, shown in black, is not live. The monitors, $\{c_1, c_2\}$, and the arcs involving them, $\{(c_1, t_4), (t_5, c_1), (c_2, t_5), (t_4, c_2)\}$, are shown in red. A single token (shown in red) is placed in monitor c_2. The resulting, structurally-enhanced, PN is live, and the influence of the monitors is equivalent to a minimally restrictive LESP (cf. section 6.2 for additional details about this construction).

2.4 Review of Polyhedral Theory

The set $\mathcal{M} \in \mathcal{R}^n$ is called convex if the line segment connecting each pair of points in $\mathcal{M}$ also lies in $\mathcal{M}$. To expand this notion to integer-valued sets we have two different definitions:

- The integer set $\mathcal{M} \in \mathcal{N}^n$ is called *segmentally* convex if all the integer points on the line segment connecting each pair of points in $\mathcal{M}$ also belongs to $\mathcal{M}$.
- The integer set $\mathcal{M} \in \mathcal{N}^n$ is called *intersection* convex if there is a real convex set C such that its set of integer points equals to $\mathcal{M}$.

Although these two definitions are equivalent for real sets, they are not equivalent when it comes to integral sets. If the set is *intersection* convex, it is also *segmentally* convex but the converse is not true. Throughout this thesis, by convexity we mean *intersection* convexity.

The *convex combination* of a set of points, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k$ in both integer and real set is defined as $\sum_{i=1}^{k} \lambda_i \mathbf{x}_i$, where $\lambda_1, \lambda_2, \cdots, \lambda_k$ are non-negative real numbers such that $\sum_{i=1}^{k} \lambda_i = 1$. If the latter condition is dropped, then the combination is called *conic combination* and if only the second property holds, it is called *affine combination*. The *convex hull* of these points is the set of all possible convex combination of the points and will be shown as $conv(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k)$.

We present a *polyhedron* as $P(\mathbf{A}, \mathbf{b}) := \{\mathbf{x} \in \mathcal{R}^n | \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$ where for $n, m \in \mathcal{N}$, $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{b}$ is an $m-$dimensional vector. If the entries of both $\mathbf{A}$ and $\mathbf{b}$ are rational numbers, the polyhedron $P(\mathbf{A}, \mathbf{b})$ is called *rational polyhedron*. The set of integer points inside the polyhedron $P(\mathbf{A}, \mathbf{b})$ will be denoted by *Int*$(P(\mathbf{A}, \mathbf{b}))$. A *polytope* is the convex hull of a finite set of points in $\mathcal{R}^n$. It is easy to see that polytope is a bounded polyhedron.

The *Minkowski sum* of two sets of vectors, $A, B \subseteq \mathcal{R}^n$ is a set of all possible summation between the members of these sets, i.e, $\{\mathbf{a} + \mathbf{b} \mid \mathbf{a} \in A, \mathbf{b} \in B\}$. Also, the polyhedron $P(\mathbf{A}, \mathbf{b})$ can be defined in another way by *Affine Minkowski-Weyl Duality* theorem: A subset $P \subseteq \mathcal{R}^n$ is a polyhedron if and only if it is the Minkowski sum of a polytope and a finitely generated cone.

A *half-space* $(\mathbf{w}, t)$ is the set $\{\mathbf{x} \in \mathcal{R}^n \mid \mathbf{w}^T \mathbf{x} \geq t\}$ for $\mathbf{w} \in \mathcal{R}^n$, $t \in R$. We use the notation $\{(\mathbf{w}_i, t_i)\}_{i=1}^{k}$ to denote the intersection of a set of k-many half-spaces. A half-space $(\mathbf{w}, t)$ is a *valid inequality* for a set $\mathcal{S} \subset \mathcal{R}^n$, if $\mathcal{S} \subset \{\mathbf{x} \in \mathcal{R}^n \mid \mathbf{w}^T \mathbf{x} \geq t\}$. $\mathcal{F}$ is a *face* of the polyhedron $P(A, b)$, if $\mathcal{F} \subset P(\mathbf{A}, \mathbf{b})$ and there exists a valid inequality $(\mathbf{w}, t)$ for $P(\mathbf{A}, \mathbf{b})$ such that $\mathcal{F} = \{\mathbf{x} \in P(\mathbf{A}, \mathbf{b}) \mid \mathbf{w}^T \mathbf{x} = t\}$. If $\mathcal{F} \neq \varnothing$, then $(\mathbf{w}, t)$ *supports* the face $\mathcal{F}$, and $\mathcal{F} = \{\mathbf{x} \in P(\mathbf{A}, \mathbf{b}) \mid \mathbf{w}^T \mathbf{x} = t\}$ is called the *supporting hyperplane* of $\mathcal{F}$. A supporting hyperplane is called *right-closed supporting hyperplane* if both $\mathbf{w}$ and t are non-negative. $\mathcal{F}$ is called *proper-face* if $\mathcal{F} \neq P(\mathbf{A}, \mathbf{b})$ and *non-trivial* if $\mathcal{F} \neq \varnothing$. A *facet* of the polyhedron $P(\mathbf{A}, \mathbf{b})$, $\mathcal{F}$, is the proper face of $P(\mathbf{A}, \mathbf{b})$ such that it is not strictly contained in any proper or non-trivial face of $P(\mathbf{A}, \mathbf{b})$. A *right-closed facet* of a polyhedron $P(\mathbf{A}, \mathbf{b})$ is a subset of a right-closed supporting hyperplane.

2.5 Convex Hull

Since computing the convex hull will be a fundamental part of this dissertation, we devote this section to the notion of convex hull and its computation. We will review the definition of the convex hull and then discuss different methods of computing it and their complexities. For more in depth review of these materials one can refer to references [31] and [32].

The *convex hull* of a set of points is the set of all possible convex combination of the points. Essentially, the convex hull of a set of points, $\mathcal{M} \in \mathcal{R}^n$ is the smallest convex set in $\mathcal{R}^n$ that contains the set $\mathcal{M}$. The problem of computing the convex hull is to find the representation of $conv(\mathcal{M})$ for the set $\mathcal{M}$. This can be done by enumerating the vertices or *extreme points* of the $conv(\mathcal{M})$, which is called the *vertex enumeration problem.* If $conv(\mathcal{M})$ is full dimensional, n-polyhedron $P(\mathbf{A}, \mathbf{b})$, then the problem reduces to finding proper $\mathbf{A}$ and $\mathbf{b}$, and each row of $\mathbf{A}$, along with the corresponding row of $\mathbf{b}$ will form one *facet* of $conv(\mathcal{M})$. This is called *facet enumeration problem.* As a side note, a polyhedron is full dimensional if there is an interior point $\mathbf{x} \in conv(\mathcal{M})$ that satisfies all the inequalities strictly.

2.5.1 Vertex and Facet Enumeration problems

The foundation to these two problems is the *Minkowski- Weyl* theorem for convex polyhedra which can be stated as following:

Theorem 2.5.1. *For a convex polyhedron $\mathcal{M} \subset \mathcal{R}^n$ the following statements are equal:*

- $\mathcal{M} = P(\boldsymbol{A}, \boldsymbol{b}) := \{\boldsymbol{x} \in \mathcal{R}^n | \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}\}$
- *For a finite set of vertices, $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$ and finite set of vectors, $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_l$, $\mathcal{M}$ is defined as:*

$$\mathcal{M} = conv(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k) + cone(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_l)$$

where $cone(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_l)$ is the conic combination, defined earlier.

Minkowski-Weyl theorem suggests that every convex polyhedron has two representations: vertex representation, *V-representation* and halfspace representation, *H-representation*. The problem of converting the H-representation to V-representation is what is called the *vertex enumeration* and the reverse process is the *facet enumeration* problem. If the polyhedron $\mathcal{M}$ is full dimensional and has at least one vertex, then each representation is unique, for positive multipliers of the inequalities and $\mathbf{r}_i$. Although a very fundamental problem in computational geometry, existence of a polynomial time algorithm for converting these two representation to each other is still an open question. It is interesting to know that, although facet enumeration and vertex enumeration are equivalent, for some classes of convex polyhedra, one of these problems is much easier than the other.

2.5.2 Convex Hull Algorithms

In this section, we review common convex hull algorithms and discuss their complexities in number of points and dimension. The standard practice is to consider the dimension to be fixed, because there is no good algorithm for computing a convex hull if the dimension is permitted to change.

Brute Force Algorithm

This is the simplest of the algorithms and primarily is used in a 2-dimensional space. The algorithm simply tests each line segment between two pairs of points and checks if all the other points lie in one side of this line. With an input of n points, there will be $O(n^2)$ lines along with $O(n)$ points to check whether that line is an edge, which yields an $O(n^3)$ procedure. Using a similar argument, this algorithm yields an $O(n^{d+1})$ for a problem in $d-$dimensional space, which is not satisfactory. The figure 2.4 shows the main idea behind this algorithm.

Gift Wrapping or Jarvis's March Algorithm

This algorithm sorts the given points using one of the variants of the class of sorting algorithms. After sorting the points lexicographically, or at least by one of its elements, it selects, let us say, the point that is farthest left. The idea is to find the vertices of the convex hull at each iteration, rotating either

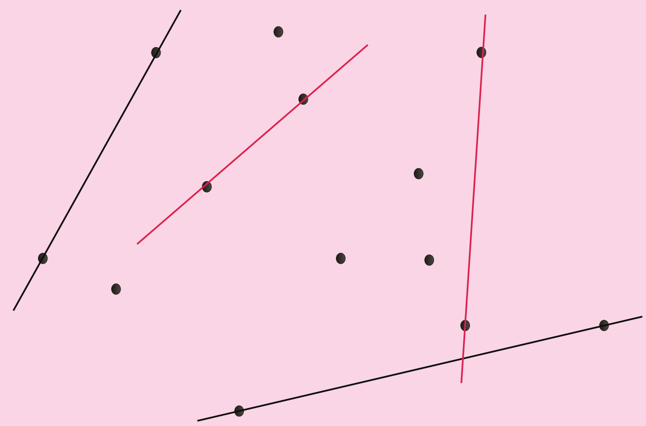


Figure 2.4: Brute Force Algorithm: The red edges are not acceptable as points are distributed at each side of them; On the other hand, black lines are good candidates for the edges of the convex hull

clockwise or counterclockwise. Let points $\mathbf{m}_{k-1}$ and $\mathbf{m}_k$ are the last points added to the convex hull computation. We connect all the remaining points to the point $\mathbf{m}_k$ and will pick the one that generates the largest angle with the $\mathbf{m}_{k-1}\mathbf{m}_k$ line segment. The step of adding new point to the convex hull can be completed in $O(n)$ time. The essential question then becomes how many vertices the convex hull will have. As the *worst case* scenario, all the given points will be a vertices of the convex hull. Hence, the algorithm in worst case scenario, will take $O(n^2)$ time. Notice that this algorithm is much easier again to perform in a 2-dimensional space, rather than in general dimension. Figure 2.5 illustrates each step of this algorithm.

Divide-and-Conquer Algorithm

The idea of this algorithm, which uses the *divide-and-conquer* design, is to compute the convex hull for the subsets of the main set and the merge these two set together. The algorithm is recursive in nature and we will discuss the complexity in more depth. The algorithm is presented in algorithm 1:

The merging algorithm mentioned in algorithm 1, is essentially uses the tangent line between two smaller convex hull, $conv(\mathcal{M}^1)$ and $conv(\mathcal{M}^2)$. For $\mathbf{m}^1 \in conv(\mathcal{M}^1)$ and $\mathbf{m}^2 \in conv(\mathcal{M}^2)$, observe that

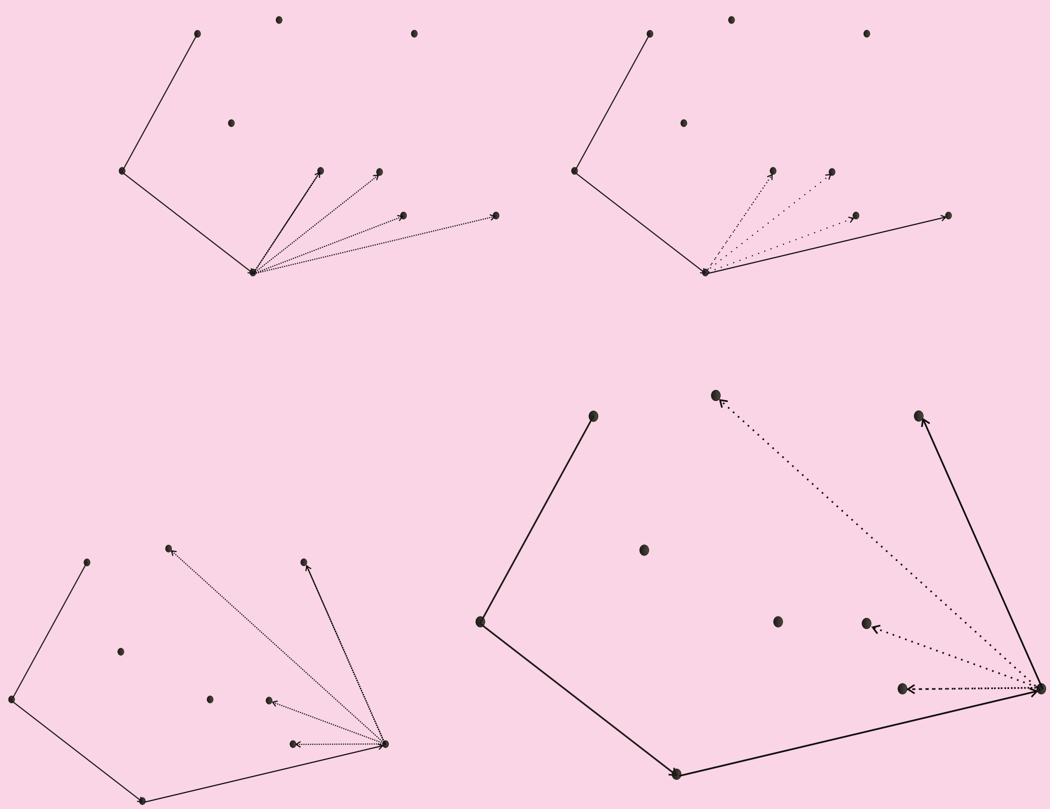
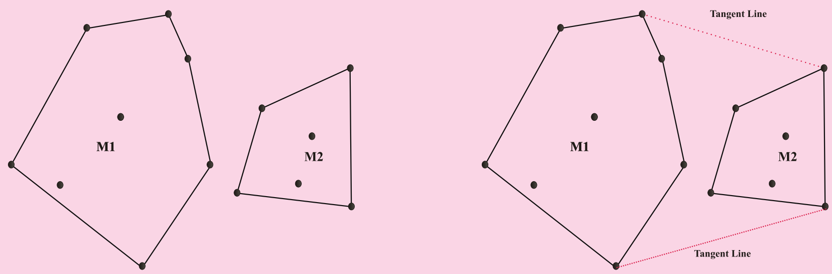


Figure 2.5: Figures (a)-(d) show the steps involved in the Gift Wrapping algorithm. At each point, an edge will be chosen if it can make the largest angle with the previous edge.

Algorithm 1 Divide-and-Conquer Algorithm

1: **if** Number of input points, $|\mathcal{M}| \leq 3$, **then**
2: Use brute force algorithm.
3: **if** $|\mathcal{M}| > 3$ **then**
4: Sort the points based on their first element or lexicographically.
5: Divide the ordered set of points in $\mathcal{M}$ to two distinct set, $\mathcal{M}^1$ and $\mathcal{M}^2$, where the first one has the firs half of the ordered point and the second one contains the rest.
6: Compute the $conv(\mathcal{M}^1)$ and $conv(\mathcal{M}^2)$ in recursive procedure.
7: Merge the $conv(\mathcal{M}^1)$ and $conv(\mathcal{M}^2)$ together using the merge algorithm
8: Exit.



(a) Divide-and-Conquer (b) Divide-and-Conquer

Figure 2.6: Figure (a) shows the convex hull of two smaller subsets; Figure (b) is the merging step, using the tangent algorithm

line $\mathbf{m}^1\mathbf{m}^2$ is *tangent*, if all the points on both hulls are located on one side of this line. The tangent algorithm is described in algorithm 2 and figure 2.7

Algorithm 2 Tangent Algorithm

Find the edge $\mathbf{m}^1\mathbf{m}^2$, that does not intersect two hulls, where $\mathbf{m}^1 \in conv(\mathcal{M}^1)$ and $\mathbf{m}^2 \in conv(\mathcal{M}^2)$
while $\mathbf{m}^1\mathbf{m}^2$ is not the tangent line for both hulls **do**
    while $\mathbf{m}^1\mathbf{m}^2$ is not tangent for $conv(\mathcal{M}^1)$ **do**
        Move $\mathbf{m}^1$ clockwise
    while $\mathbf{m}^1\mathbf{m}^2$ is not tangent for $conv(\mathcal{M}^2)$ **do**
        Move $\mathbf{m}^2$ counterclockwise
Exit.

Algorithm 1, uses the brute force method if the input size is less than 3 which can be completed in $O(1)$ time. Therefore, using the simple divide-and-conquer time analysis we have:

$$T(n) = \begin{cases} O(1) & \text{if } |\mathcal{M}| \leq 3 \\ n + 2T(\frac{n}{2}) & \text{if } |\mathcal{M}| = n > 3. \end{cases}$$

Note that the tangent algorithm can be performed in $O(n)$ time. Therefore, the time complexity for divide-and-conquer algorithm is $T(n) = O(n \log n)$.

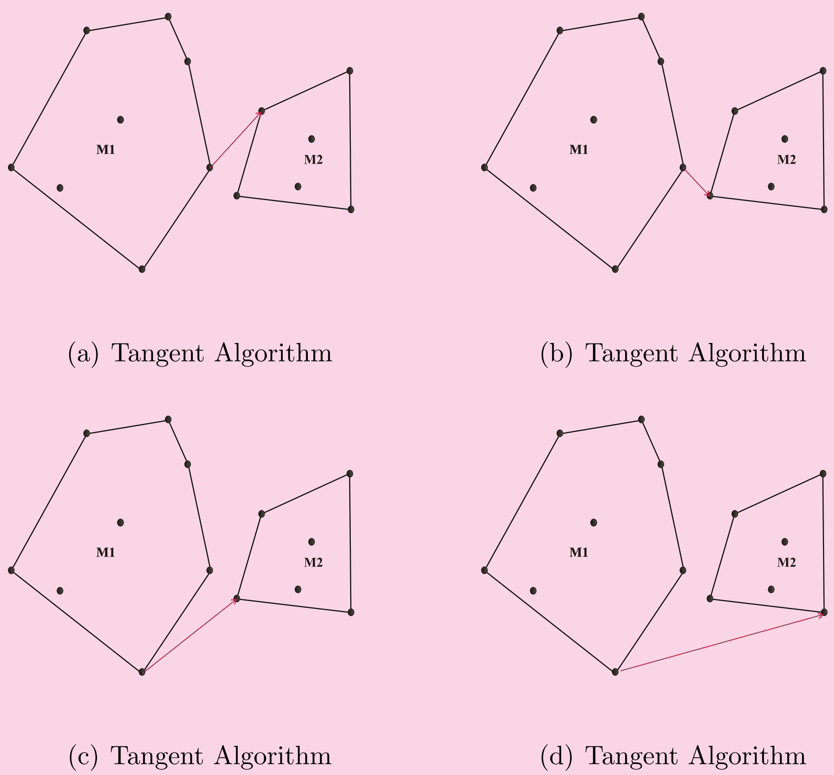


Figure 2.7: Steps of finding the tangent line for merging step

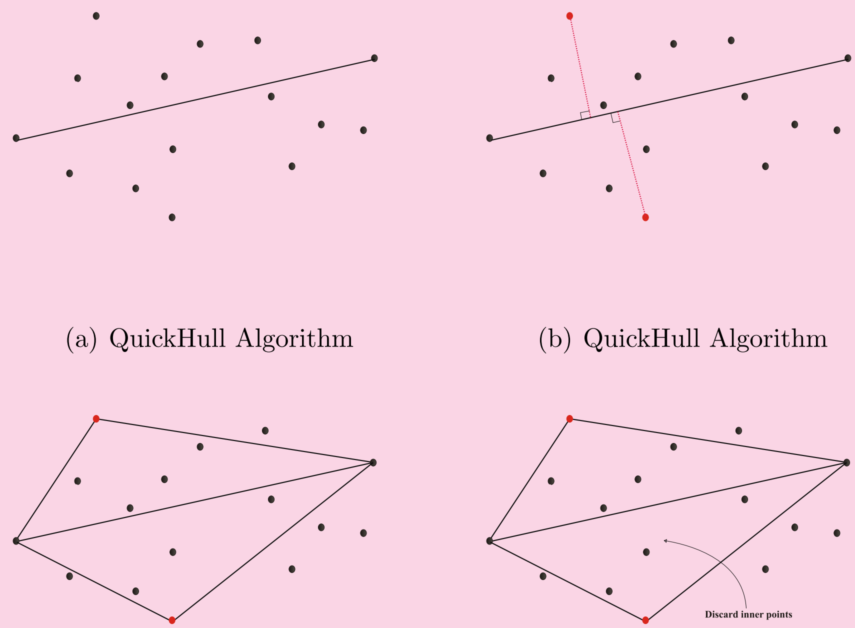
QuickHull Algorithm

The *QuickHull* algorithm is one of the widely used convex hull algorithms in practice. In this dissertation, we use QuickHull algorithm for computing the convex hull. The name of the algorithm comes from the fact that it uses the divide-and-conquer approach used in the *quicksort* algorithm. The idea behind this algorithm is to construct the convex hull over the small subsets of the main set and then eliminate all the interior points. We describe the steps of the algorithm, for a 2-dimensional space, as following:

Algorithm 3 QuickHull Algorithm

Sort the points by the first element.
while set of inner points are not $\varnothing$ **do**
    draw a line between the first and last point in the ordered set, $\mathbf{m}^1$ and $\mathbf{m}^1$.
    Find two points that have the maximum perpendicular distance form $\mathbf{m}^1\mathbf{m}^2$ for each side.
    construct the two triangle(convex hull) of $conv(\mathcal{M}^1)$ and $conv(\mathcal{M}^2)$ by these 4 points.
    Discard the points inside the $conv(\mathcal{M}^1)$ and $conv(\mathcal{M}^2)$
    Return to step 5 and do it over the remaining points
Exit.

Eliminating the inner point can be a challenging problem, specially when we are dealing with problems in high dimension. This is is comparatively easy for problems in two dimensions. this can be easy. We need to find the point that produces the largest perpendicular distance from the separating line and then, forming a triangle over the separating line. All the inner points inside the triangle then will be removed. We repeat the same process with each edge of the formed triangle: making new edges and removing the inner edges. The computational complexity of the QuickHull algorithm heavily depends on how the input points are distributed over the plane. If the number of points $card(\mathcal{M}) = n$, then for each element of the points, the maximum and minimum can be found $O(n)$. Using only the first element, can be a good start and then set can be split in half. Assuming that points are evenly spread



(a) QuickHull Algorithm

(b) QuickHull Algorithm

(c) QuickHull Algorithm

(d) QuickHull Algorithm

Figure 2.8: Steps of QuickHull Algorithm

in two side of the separating line, we can get:

$$T(n) = \begin{cases} O(1) & \text{if } card(\mathcal{M}) = 1 \\ 2T(\frac{n}{2}) & \text{if } card(\mathcal{M}) = n > 1. \end{cases}$$

which gives us the computational time of $T(n) = O(n \log n)$. The *worst case scenario* happens when the points are not distributed evenly, which makes it hard to estimate the complexity time of the algorithm. It has been proved that the algorithm performs in $T(n) = O(n^2)$ time, when in the worst case possible. In practice this algorithm performs well on average, and therefore we will use this algorithm.

In this dissertation, we deal with higher dimensional problems which simple form of the mentioned algorithms cannot be used. We will utilize the software tool *polymake* developed in [33, 34]. Also for general complexity of a convex hull computation algorithm, for a fixed dimension and specifically a deterministic version of the algorithm, one can refer to theorem 4.2.4 and reference [35].

2.6 Discrete Volume

The *discrete volume* of a geometric set, $\mathcal{M}$, is the number of lattice points inside the set or simply $\mathcal{M} \cap \mathcal{Z}^n$, where $\mathcal{Z}^n$ is the set of all integers in a n-dimensional Euclidean space. The discrete volume can be computed by:

$$vol(\mathcal{M}) = \lim_{t \to \infty} Int(\mathcal{M} \cap \frac{1}{t}\mathcal{Z}^n)\frac{1}{t^n}$$

where it calculates the number of lattice points inside the geometrics set $\mathcal{M}$ where it has been shrunk by a factor of t. The total volume, comes as a natural integration over the factor t. Unfortunately the equation above is not easy to calculate for any geometry and can become a hard problem to solve in an arbitrary setting. A detailed treatment can be found in reference [36]. In this section we will focus on counting the lattice points inside an n-simplex, specially the *standard simplex*, which is used to analyze the performance of a grid-search based algorithm in Chapter 5.

Before jumping to counting the lattice points, we will discuss the *generating functions* in brief, as they are fundamental to the issue of computing the discrete volume of polyhedra.

2.6.1 Generating Functions

Herbert Wilf says in his book [37].

> " *A generating function is a clothesline on which we hang up a sequence of numbers for display* ."

Essentially what it means is that if we have a sequence of numbers a_n, which are indexed by natural numbers, by using generating functions, we can get more information about the sequence. Most of the time, solving the sequence, finding the closed form solution for the sequence, will be impossible. Using the *power series*, whose coefficients are the members of the sequence, is an standard tool to tackle this problem. Solving a sequence in general, aims to find either an exact closed form solution or a recursive formula for the members of sequence.

For the sequence of numbers, $\langle a_n \rangle = a_0, a_1, \ldots$, the generating function $A(x)$ is defined as the formal power series:

$$A(x) = \sum_{i=0}^{\infty} a_i x^i$$

where the $\langle a_n \rangle$ are the coefficients of this series. This can be shown by the following notation: $[x^n]A(x)$. Then, the problem reduces to find the $[x^n]A(x)$ for the defined power series. Note that generating functions are not functions in their traditional definition and usually called *generating series.* The following example will show how these series can become useful, when we are dealing with sequences.

Example

Let consider the *Fibonacci Numbers*, where each term is the sum of two previous terms:

$$F_{n+2} = F_{n+1} + F_n$$

We can write down the value of F_n when n is small, but when n is large, we will need an exact formula.

The generating function is particularly useful in this regard. We define the following generating function:

$$\mathbf{A}(x) = \sum_{n=0}^{\infty} F_n x^n$$

Plugging the formula for the sequence in the generating function above, we can write:

$$\sum_{n=0}^{\infty} F_{n+2} x^n = \sum_{n=0}^{\infty} F_{n+1} x^n + \sum_{n=0}^{\infty} F_n x^n$$

The left-hand side of the equation contains terms that are not presented on the right-hand side. So, as a general technique in solving such a problem, we can expand left-hand side and write:

$$\sum_{n=0}^{\infty} F_{n+2} x^n = \frac{1}{x^2} \sum_{n=2}^{\infty} F_n x^n = \frac{1}{x^2}(\mathbf{A}(x) - x)$$

where we know that the first two terms of the Fibonacci series are 0 and 1. We just take these two numbers out in order to match the right-hand side. Applying the same technique for the right-hand side, then multiplying both sides of the equality by x^n, and summing over $n \geq 1$ we have:

$$\frac{1}{x^2}(\mathbf{A}(x) - x) = \frac{1}{x}\mathbf{A}(x) + \mathbf{A}(x)$$

Collecting the term $\mathbf{A}(x)$ from both sides yields:

$$\mathbf{A}(x) = \frac{x}{1 - x - x^2}$$

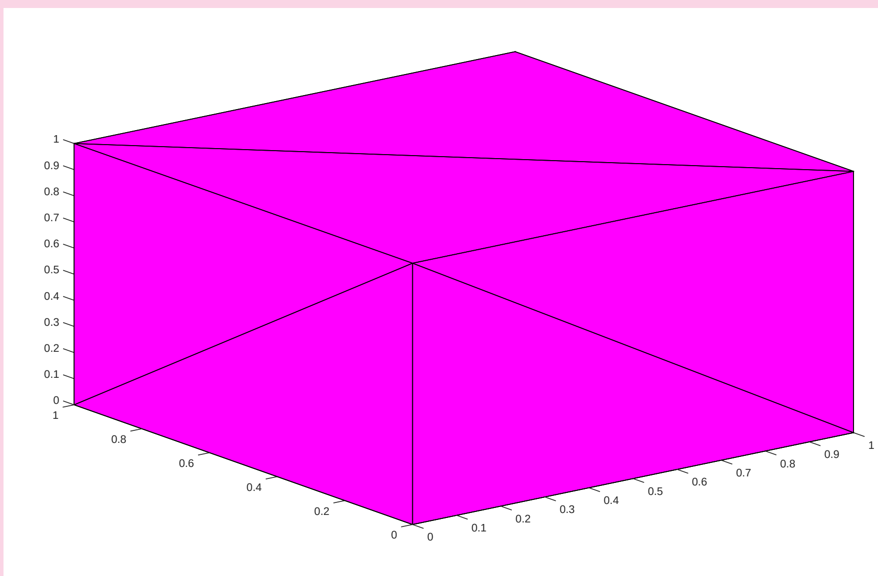
Using *partial fraction decomposition* and results in *geometric series* convergence of $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$, we can find a closed form solution for each term of Fibonacci numbers:

$$[x^n]A(x) = \mathbf{F}_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

In many enumeration problems, generating functions are used as the fundamental technique to recover the solution. As we discuss in the next chapter, for counting the lattice points inside the polyhedron, we will use a very well-known generating function, called *Ehrhart series.*

2.6.2 Lattice Points in Simplex

This subsection starts with a simple example before discussing the main problem. An square with a length k, will have $(k+1)^2$ lattice points, given the fact that all the vertices are integral points. Expanding this idea to a cube in 3-D will yield a result of $(k+1)^3$. Now take a unit cube, in a n-dimensional space, where all for all the vertices $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$, each $\mathbf{v}_i$ is either 0 or 1. By expanding the cube by factor of t or *dilate* it by t, the resulting cube will have $(t+1)^n$ lattice points overall, out of which, $(t-1)^n$ of them will be in the interior of the dilated cube.



(a) Unit Cube

Figure 2.9: A unit cube, where the dilation factor $t = 1$; This means that it has $(1+1)^3 = 8$ lattice points

For any geometric set $\mathcal{M} \in \mathcal{R}^n$, we define *Ehrhart polynomial* or *Lattice-point enumerator* function as:

$$L_{\mathcal{M}}(t) = Int(\mathcal{M} \cap \frac{1}{t}\mathcal{Z}^n)$$

which essentially is the discrete volume of the dilated set (by factor of t), defined previously. As we discussed earlier, this term is easy to recover for the unit cube which is $L_{\mathcal{M}}(t) = (t+1)^n$, when $\mathcal{M}$ is a unit cube. For complex geometries, this would not be easy. This calls for the use of generating functions. The *Ehrhart Series* of bounded geometry $\mathcal{M}$ is defined as the generating function over $L_{\mathcal{M}}(t)$:

$$Ehr_{\mathcal{M}}(x) := 1 + \sum_{t=1}^{\infty} L_{\mathcal{M}}(t)x^t$$

By replacing $L_{\mathcal{M}}(t) = (t+1)^n$, we get

$$Ehr_{\mathcal{M}}(x) = 1 + \sum_{t=1}^{\infty}(t+1)^n = \sum_{t=0}^{\infty}(t+1)^n$$

Therefore, by applying the similar technique used in the solving the Fibonacci series problem, we can get a closed form solution for the problem:

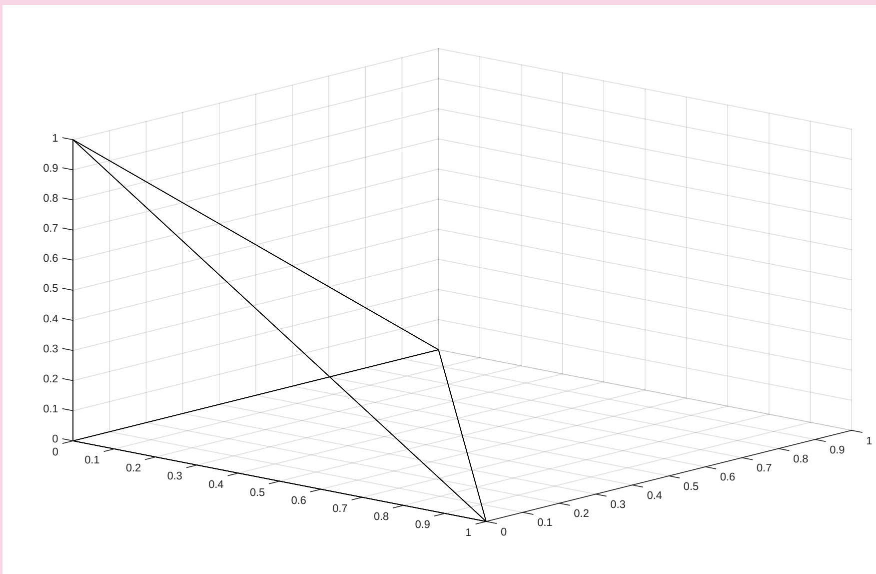
$$Ehr_{\mathcal{M}}(x) = \frac{1}{(1-x)^{n+1}} \sum_{i=1}^{n}(\sum_{j=0}^{i}(-1)^j \binom{n+1}{j}(i-j)^n)x^{i-1}$$

Although, in this case the lattice-point enumerator looks much easier than the Ehrhart series, but this will be absolutely critical for more complex geometries. Now we can focus on the *simplex*, which is essential to this dissertation.

An n-dimensional polytope with exactly $n+1$ vertices is called $n-$*simplex*. A polytope has dimension n, if the dimension of its affine spaces is n. A standard simples $\mathcal{M}$ is a convex hull of $n+1$ vertices: n unit vectors and origin.

The hyperplane representation of the $\mathcal{M}$ can be written as:

$$\mathcal{M} = \{\mathbf{x} \mid \mathbf{x} \in \mathcal{R}^n \text{ and } \mathbf{1}^T \cdot \mathbf{x} \leq 1 \text{ and } \mathbf{x} \geq 0\}$$



(a) Simple Simplex

Figure 2.10: Simple 3-simplex

where **1** is the vector of all ones. The number of lattice points in the simple simplex is similar to very famous problem: number of integral solutions to a linear equality. For the dilation of the simplex by factor of t, we can add an *slack* variable, and then problem becomes:

$$\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_n + \mathbf{x}_{n+1} = t$$

Number of integer solution to the equation above is equal to the number of lattice points in a dilated simple simplex. So, if p_n is the number of integral solutions to the equation above when the right-hand side is n, by using the generating function used in *Frobenius problem* and *Coin Exchange Problem*, we can define the following generating function for the mentioned counting problem:

$$\mathbf{A}(z) = \sum_{n=0}^{\infty} p_n z^n = \sum_{\mathbf{x}_1=0}^{\infty} z^{\mathbf{x}_1} \sum_{\mathbf{x}_2=0}^{\infty} z^{\mathbf{x}_2} \cdots \sum_{\mathbf{x}_{n+1}=0}^{\infty} z^{\mathbf{x}_{n+1}}$$

By using the fact that: $\sum_{k\geq 0} z^k = \frac{1}{1-z}$, and multiplying the right-hand side by z^{-t}, we can write:

$$\mathbf{A}(z) = z^{-t} \frac{1}{(1-z)(1-z)\cdots(1-z)} = \frac{1}{(1-z)^{n+1} z^t} \tag{2.1}$$

Hence, the p_n will be the constant term of $\mathbf{A}(z)$, $p_n = [z^n]\mathbf{A}(z)$. Note that generating series $\mathbf{A}(z)$ is the Ehrhart polynomial of the simple simplex. In order to recover the constant terms of such a series, *binomial series*, can be beneficial. A general binomial series can be presented as: $(1+x)^n = \sum_{k=0}^{\infty} \binom{n}{k} x^k$. By plugging in the $-z$ and $-t$, we have:

$$\frac{1}{(1-z)^{n+1}} = \sum_{k=0}^{\infty} \binom{-(n+1)}{k} (-z)^k$$

where by a simple expansion, we can find out that:

$$\binom{-(n+1)}{k} = (-1)^k \binom{k+n}{k}$$

which means that:

$$\frac{1}{(1-z)^{n+1}} = \sum_{k=0}^{\infty} \binom{n+k}{n} z^k \tag{2.2}$$

So the constant term of equation 2.1 is the same as coefficients in equation 2.2. Therefore, the number of lattice points in a simple simplex, dilated by factor of t will be:

$$L_{\mathcal{M}}(t) = \binom{n+t}{t} \tag{2.3}$$

So, we can summarize all the discussion above with the following theorem.

Theorem 2.6.1. *For a simple simplex $\mathcal{M}$ in an n-dimensional space, dilated by factor of t, we have:*

- *Lattice point enumerator function or Ehrhart polynomial will be:* $L_{\mathcal{M}}(t) = \binom{n+t}{t}$
- *The Ehrhart series is defined as:* $Ehr_{\mathcal{M}}(z) = \frac{1}{(1-z)^{n+1}}$

To conclude this section, it is worth mentioning that equation 2.3, gives all the lattice points in the simplex $\mathcal{M}$, even the ones on the boundary of the polyhedron. To compute the points which locate inside the boundaries of the simplex, Computing lattice points in the interior of such a geometry involves more steps as one needs to eliminate the integral points on the facets of the simplex. The closed form solution for the interior lattice-enumerator can be achieved by simply evaluating the $L_{\mathcal{M}}(t)$ at the negative integers which is $(-1)^n L_{\mathcal{M}}(-t)$. Additional information can be found in reference [36].

Chapter 3

Right-closed Integral sets and the Liveness Problem

This chapter serves as the motivation behind exploring the properties of the right-closed sets and in specific, the convexity of these sets. First, a short background is presented regarding the issues with computing the minimally restrictive LESPs. Then preliminary results regarding specific PNs are discussed, which show the importance of right-closed sets in liveness problem.

3.1 On Existence of LESPs for certain PN Models

It has been shown that the existence of an LESP for an arbitrary PN is *undecidable* [20], meaning there is not a single algorithm which can answer whether or not an LESP exists for an arbitrary PN. More interestingly, both existence and the non-existence of an LESP for an arbitrary PN are not *semi-decidable* problems. These two arguments leads to the conclusion that every single heuristic algorithm for finding an LESP for an arbitrary PN will not terminate in at least one instance where there is an LESP, and at least in one instance which there is no LESP. Although, if there exists an LESP for an arbitrary PN, then there should be a unique minimally restrictive LESP for that PN.

On the bright side, for some classes of PNs, including ordinary *Free-Choice*PNs (FCPNs), and PNs where all the transitions are controllable, the minimally restrictive LESP, when it exists is *marking monotone*. That is, if a transition is control-enabled at some marking by a *marking monotone* policy, it remains control-enabled for all larger markings. A PN structure $N = (\Pi, T, \Phi, \Gamma)$ is *Free-Choice* if $\forall p \in \Pi$, $card(p^\bullet > 1) \Rightarrow {}^\bullet(p^\bullet) = \{p\}$, where $card(\bullet)$ denotes the cardinality of the set argument. FCPNs are widely used to model the flow of material in manufacturing/logistics systems, and the flow of control in data-flow computation.

We say the set of markings $\mathcal{M} \in \mathcal{N}^n$ is *control-invariant* with respect to a partially controlled PN structure $N = (\Pi, T, \Phi)$ if $\mathcal{M} = \Gamma(\mathcal{M})$, where $\Gamma(\mathcal{M}) = \{\mathbf{m}^i \in \mathcal{N}^n \mid \exists \sigma \in T_u^*, \exists \mathbf{m}^j \in \mathcal{M}$ such that $\mathbf{m}^j \xrightarrow{\sigma} \mathbf{m}^i\}$. In general, $\mathcal{M} \subseteq \Gamma(\mathcal{M})$. What we mean by control invariant is that firing an uncontrollable transition will not produce a marking which does not belong to $\mathcal{M}$ and only firing a controllable transition can result in marking out of $\mathcal{M}$.

Let define the set $\Delta(N)$ for every PN as following:

$$\Delta(N) : \{\mathbf{m}^0 \mid \exists \text{ an LESP for } N(\mathbf{m}^0)\} \tag{3.1}$$

This is the set of initial marking for which there is an LESP for a PN structure N. It has been shown that $\Delta(N)$ is control invariant with respect to PN structure N, meaning if $\mathbf{m}^1 \in \Delta(N)$, $t_u \in T_e(N, \mathbf{m}^1) \cap T_u$, and $\mathbf{m}^1 \xrightarrow{t_u} \mathbf{m}^2$ in N, then $\mathbf{m}^2 \in \Delta(N)$. The following lemma utilizes this property to re-define a minimally restrictive LESP.

Lemma 3.1.1. *[20] The supervisory policy that control-disables a transition only if its firing at a marking in $\Delta(N)$ would result in a new marking that is not in $\Delta(N)$, is the minimally restrictive LESP for $N(\mathbf{m}^0)$ for $\mathbf{m}^0 \in \Delta(N)$*

For a PN N, if any initial marking $\mathbf{m}^0$ belongs to $\Delta(N)$, then it is guaranteed to have an LESP. Therefore, the problem of synthesizing the minimally restrictive LESP translates to problem of computing the $\Delta(N)$-set. Unfortunately $\Delta(N)$ is not computable in general. The following result from the same reference [20], shows that membership problem to set $\Delta(N)$ is semi-decidable.

Lemma 3.1.2. *For an arbitrary PN $N(\mathbf{m}^0)$, and for an integral vector $\mathbf{m} \in \mathcal{N}^n$, neither "$\mathbf{m} \in \Delta(N)$?" nor "$\mathbf{m}^0 \notin \Delta(N)$?" is semi-decidable.*

As a result, any heuristic algorithm for computing a LESP will hang indefinitely for at least one instance where there is LESP and once for the case which there is not one. This leaves us with the only option of finding PNs for which the $\Delta(N)$-set has a specific structure that permits its computation.

3.2 Right-Closed $\Delta(N)$

As discussed above, for some classes of PNs when the minimally restrictive LESP exists, it is *marking monotone*. This observation gives rise to exploring the PNs for which the corresponding $\Delta(N)$ is right-closed. The following two results from [20] resolve the computability issue for $\Delta(N)$.

Theorem 3.2.1. *If N is any PN structure with all its transitions controllable (i.e $T_c = T$), then the finite set* $\min \Delta(N)$ *is computable.*

This result is particularly interesting as for any PN, $\Delta(N) \subseteq \Delta_f(N)$ where $\Delta_f(N)$ is the set of initial marking for which there exists an LESP where all the transitions are controllable.

Theorem 3.2.2. *The largest subset of any right-closed set of marking $\mathcal{M} \subseteq \mathcal{N}^n$, that is control-invariant with respect to a PN structure N can be computed.*

Based on the aforementioned theorems, a feasible process can be built in order to synthesize the minimally restrictive LESP for a PN.

3.2.1 Synthesizing $\Delta(N)$

Suppose (1)N is a PN structure where $\Delta(N)$ is known to be right-closed, (2)Ψ is a set of markings that are control invariant with respect to N and it is right-closed set, (3)$\mathcal{P}_\Psi$ is a supervisory policy which control-disables any controllable transition at a marking which belongs to Ψ if its firing would result in a marking which is not in Ψ and (4)$\mathbf{m}^0 \in \Psi$. The *coverability graph* of $N(\mathbf{m}^0)$ under supervision of $\mathcal{P}_\Psi$ is denoted by $G(N(\mathbf{m}^0), \mathcal{P}_\Psi)$. Then the policy $\mathcal{P}_\Psi$ is a liveness enforcing policy if and only if:

1. $\mathbf{m}^0 \in \Psi$ and

2. *Path-requirement*: There is a vertex v, and a closed-path $v \xrightarrow{\sigma} v$ in $G(N(\mathbf{m}^i), \mathcal{P}_\Psi)$, where $\sigma \in T^*$, for each $\mathbf{m}^i \in min(\Psi)$, where

   (a) all transitions appear at least once in $\sigma (i.e. \mathbf{x}(\sigma) \geq 1)$, and

(b) the net-change in the token-load in each place after the firing of σ is non-negative ($i.e.\mathbf{Cx} \geq 0$).

The conditions mentioned above can be checked easily with an *Integer Linear Program* (ILP). The algorithm for synthesis of an LESP for a PN structure N that belongs to a class where $\Delta(N)$ is known to be right-closed essentially involves a search for a right-closed set of marking Ψ that is control invariant with respect to N, where each member of $min(\Psi)$ meets the path-requirement on its coverability graph described above. This is done in an interactive manner starting with an initial set:

$$\Psi_0 = \{\mathbf{m}^0 \mid \exists \text{ an LESP for } N(\mathbf{m}^0) \text{ if all transitions in } N \text{ are controllable}\}$$

which is known to be right-closed. If Ψ is the largest subset that meets the aforementioned requirement, $min(\Psi)$, effectively represents the minimally restrictive LESP for the PN $N(\mathbf{m}^0)$.

By discussion above, we have an structured method to compute the $\Delta(N)$, for specific class of PNs, where $\Delta(N)$ is known to be right-closed. One can draw the direct comparison between $\Delta(N)$ and Ψ The following theorem is the result of the mentioned procedure.

Theorem 3.2.3. *If N is a PN structure such that $\Delta(N)$ is right-closed, then problem of whether or not $\mathbf{m}^0 \in \Delta(N)$, is decidable.*

3.2.2 PNs with Right-Closed $\Delta(N)$

Theorem 3.2.3, brings about a considerable attention towards different calsses of PNs, such that the corresponding $\Delta(N)$ is known to be right-closed. In this section, the preliminary results for such classes of PNs is discussed.

Let $\Omega(t) = \{\widehat{t} \in T | {}^\bullet t \cap {}^\bullet \widehat{t} \neq \varnothing\}$, denote the set of transitions that share a common input place with $t \in T$ for a PN structure N. Let $\widetilde{\mathcal{H}}$ denote a class of PN structures where the following property is true:

$$\forall \mathbf{m} \in \Delta(N), \forall t_u \in T_u, \forall t \in \Omega(t_u), (t \in T_e(N, \mathbf{m})) \Rightarrow (t_u \in T_e(N, \mathbf{m})) \tag{3.2}$$

That is, if a transition t, a member of the $\widetilde{\mathcal{H}}$ class of PNs, is state enabled at a marking in $\Delta(N)$, then all uncontrollable transitions that share a common input place with t are also state-enabled at the same marking. The following lemma finds use in the proof of theorem coming theorems.

Lemma 3.2.4. *Let $\mathcal{P}$ be a LESP for $N(\boldsymbol{m}^0)$, where $\boldsymbol{m} \in \Delta(N)$, for a PN structure $N \in \widetilde{\mathcal{H}}$. Suppose $\boldsymbol{m}^0 \xrightarrow{\sigma} \boldsymbol{m}^i$ under the supervision of $\mathcal{P}$, and $\widehat{\boldsymbol{m}}^0 \xrightarrow{\widehat{\sigma}} \widehat{\boldsymbol{m}}^j$ without supervision in N, where the number of occurrence of each controllable transition in σ and $\widehat{\sigma}$ are identical, and $\widehat{\boldsymbol{m}}^0 \geq \boldsymbol{m}^0$. There exists strings $\sigma_1, \widehat{\sigma}_1 \in T^*$ such that:*

1. *$\boldsymbol{m}^0 \xrightarrow{\sigma\sigma_1} \boldsymbol{m}^k$ under the supervision of $\mathcal{P}$ in N.*
2. *$\widehat{\boldsymbol{m}}^0 \xrightarrow{\widehat{\sigma}\widehat{\sigma}_1} \widehat{\boldsymbol{m}}^l$ without supervision in N, and*
3. *$\boldsymbol{x}(\sigma\sigma_1) = \boldsymbol{x}(\widehat{\sigma}\widehat{\sigma}_1)$, and $\widehat{\boldsymbol{m}}^l \geq \widehat{\boldsymbol{m}}^k$*

Proof. Let $\widehat{T}_u \subseteq T_u$ denote the set of uncontrollable transitions that appear more often in $\widehat{\sigma}$ as compared to σ. If $\widehat{T} = \varnothing$, then $\widehat{\sigma} = \sigma$ and the result holds trivially. If $\widehat{T}_u \neq \varnothing$, there is a string σ_1 such that $\mathbf{m}^i \xrightarrow{\sigma_1} \mathbf{m}^{i+1}$ under the supervision of the LESP $\mathcal{P}$ such that:

1. at least one member of $t_u \in \widehat{T}_u$ is state-enabled at $\mathbf{m}^{i+1}$, and
2. none of the members of $\widehat{T}_u$ are state-enabled at any marking that results from the firing of a proper prefix of σ_1 at $\mathbf{m}^i$

It follows that $\widehat{\mathbf{m}}^j \xrightarrow{\widehat{\sigma}_1} \widehat{\mathbf{m}}^{j+1}$, without any supervision, in N. If this were not the case, there must be a proper prefix of σ_1 of the format $\overline{\sigma} t_m$, such that $\widehat{\mathbf{m}}^j \xrightarrow{\overline{\sigma}_1} \overline{\mathbf{m}}$ in N, but $t_m \in T_e(N, \overline{\mathbf{m}})$. Additionally, $t_m \in \Omega(\overline{t}_u)$ for some $\overline{t}_u \in \widehat{T}$. Since $N \in \widetilde{\mathcal{H}}$, and $\overline{\mathbf{m}} \in \Delta N$, it follows that $t_u \in T_e(N, \overline{\mathbf{m}})$, which contradicts the second requirement.

Suppose $\mathbf{m}^i \xrightarrow{\sigma_1 t_u} \mathbf{m}^{j+1}$ under $\mathcal{P}$ in N, and $\widehat{\mathbf{m}}^j \xrightarrow{\widehat{\sigma}_1} \widehat{\mathbf{m}}^{j+1}$ without supervision in N. We let $\mathbf{m}^j \leftarrow \mathbf{m}^{j+1}$, $\widehat{\mathbf{m}}^j \leftarrow \widehat{\mathbf{m}}^{j+1}$, $\sigma \leftarrow \sigma\sigma_1 t_u$, and $\widehat{\sigma} \leftarrow \sigma\sigma_1 t_u$. The result follows by repeating the above construction as often as necessary till $\widehat{T}_u = \varnothing$ □

The following theorem notes that $\Delta(N)$ is right-closed if $N \in \widetilde{\mathcal{H}}$.

Theorem 3.2.5. *[25]* $((\mathbf{m}^0 \in \Delta(N)) \wedge (\widehat{\mathbf{m}}^0 \geq \mathbf{m}^0)) \Rightarrow (\widehat{\mathbf{m}}^0 \in \Delta(N))$, *if* $N \in \widetilde{\mathcal{H}}$

Proof. Since $\mathbf{m}^0 \in \Delta(N)$, there is an LESP $\mathcal{P}$ for $N(\mathbf{m}^0)$. Following reference [20], we define an LESP $\widehat{\mathcal{P}}$ for $N(\widehat{\mathbf{m}}^0)$ as follows:

1. $\forall t \in T, \widehat{\mathcal{P}}(\widehat{\mathbf{m}}^0, t) = \mathcal{P}(\widehat{\mathbf{m}}^0, t)$

2. if $\widehat{\mathbf{m}}^0 \stackrel{\widehat{\sigma}}{\rightarrow} \widehat{\mathbf{m}}^i$ under $\widehat{P}$, then

   - $\forall t_u \in T_u, \widehat{\mathcal{P}}(\widehat{\mathbf{m}}^i, t_u) = 1$, and
   - $\forall t_c \in T_c, \widehat{\mathcal{P}}(\widehat{\mathbf{m}}^i, t_c) = 1$

   which results in$\exists \sigma \in T^*$, such that $\mathbf{m}^0 \stackrel{\sigma}{\rightarrow} \mathbf{m}^k$ under $\mathcal{P}$, and the number of occurrence of each controllable transition in σ and $\widehat{\sigma} t_c$ are identical.

If $\widehat{\mathbf{m}}^0 \stackrel{\widehat{\sigma}}{\rightarrow} \widehat{\mathbf{m}}^i$ under $\widehat{\mathcal{P}}$, then $\exists \sigma \in T^*$ such that $\mathbf{m}^0 \stackrel{\sigma}{\rightarrow} \mathbf{m}^j$ under $\mathcal{P}$, and the number of occurrence of each controllable transition in σ and $\widehat{\sigma}$ are identical. Using lemma 3.2.4, and the definition of $\widehat{\mathcal{P}}$, we know $\exists \widehat{\sigma}_1, \sigma_1 \in T^*$ such that $\widehat{\mathbf{m}}^0 \stackrel{\widehat{\sigma}\widehat{\sigma}_1}{\rightarrow} \widehat{\mathbf{m}}^{i+1}$ under $\widehat{\mathcal{P}}$, $\widehat{\mathbf{m}}^0 \stackrel{\widehat{\sigma}\widehat{\sigma}_1}{\rightarrow} \widehat{\mathbf{m}}^{j+1}$ under $\mathcal{P}$, and $\widehat{\mathbf{m}}^{i+1} \geq \widehat{\mathbf{m}}^{j+1}$. Consequently, for any $\sigma_2 \in T^*$ such that $\mathbf{m}^{j+1} \stackrel{\sigma}{\rightarrow} \mathbf{m}^{j+2}$ under $\mathcal{P}$, we have $\widehat{\mathbf{m}}^{i+1} \stackrel{\widehat{\sigma}_2}{\rightarrow} \widehat{\mathbf{m}}^{i+2}$ under $\widehat{\mathcal{P}}$ as well. Since $\mathcal{P}$ is an LESP for $N(\mathbf{m}^0)$, it follows that $\widehat{\mathbf{P}}$ is an LESP for $N(\widehat{\mathbf{m}}^0)$ □

The lemma 3.2.4 and previous theorem together imply the following theorem.

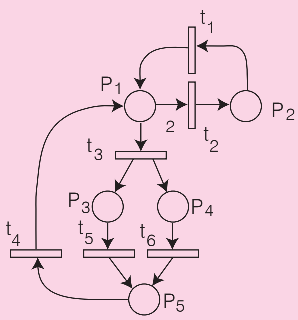
Theorem 3.2.6. ΔN *is right-closed if* $N \in \widetilde{\mathcal{H}}$

The above condition is not necessary for the right-closure of $\Delta(N)$. For instance, $\Delta(N_4)$ is right-closed for the general PN N_4 shown in figure 3.1(a), but $N_4 \notin \widetilde{\mathcal{H}}$. Specifically, $\Delta(N_4)$ is identified by the inequality $(1\ 1\ 1\ 1\ 1)\mathbf{m} \geq 1$, and $\mathbf{m} = (1\ 0\ 0\ 0\ 0)^T \in \Delta(N_4)$, $t_2 \in T_u, t_3 \in T_e(N, \mathbf{m})$, but $t_2 \notin T_e(N_1, \mathbf{m})$. The PN structure N_5 shown in figure 3.1(b) belongs to the class $\mathcal{H}$, while the PN structure N_6 of figure 3.1(c) does not.

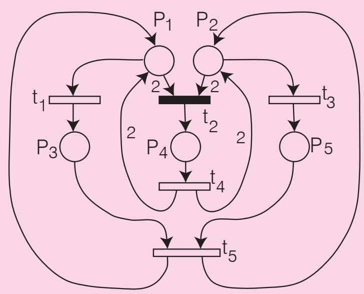
There is an LESP for the PN $N(\mathbf{m}^0)$ if and only if $\mathbf{m}^0 \in \Delta(N)$, and the existence of an LESP is undecidable for a general PN [38]. This would mean that the set $\Delta(N)$ cannot be compute for an arbitrary PN structure N. To overcome this limitation, we modify the requirement of equation 3.2 as:

$$\forall \mathbf{m} \in \mathcal{N}^n, \forall t_u \in T_u, \forall t \in \Omega(t_u), (t \in T_e(N, \mathbf{m})) \Rightarrow (t_u \in T_e(N, \mathbf{m})) \tag{3.3}$$

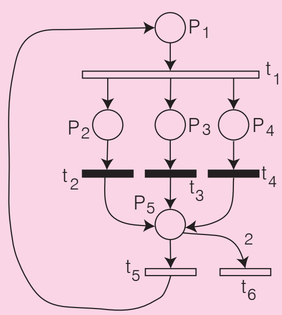
This requirement defines a class of PNs, which we denote as $\mathcal{H}(\subseteq \widetilde{\mathcal{H}})$, and from the previous theorem, we conclude that $\Delta(N)$ is right-closed for any $N \in \mathcal{H}$. The next theorem points out to the property of this newly defined class.



(a) $N_4 \notin \widetilde{\mathcal{H}}$



(b) $N_5 \in \mathcal{H}$



(c) $N_6 \notin \mathcal{H}$

Figure 3.1: (a) The PN structure N_4 is not a member of class $\widetilde{\mathcal{H}}$. However, $\Delta(N_4)$ is right-closed. (b) The PN structure N_5 is a member of $\mathcal{H}$ as it meets the structural requirements of theorem 3.2.7. (c) The PN structure N_6 is not a member of the class $\mathcal{H}$, and $\Delta(N_6) = \{\mathbf{m} \in \mathcal{N}^3 \mid (\mathbf{m}(p_1) + \mathbf{m}(p_2) + \mathbf{m}(p_3) + \mathbf{m}(p_4) \geq 1) \vee (\mathbf{m}(p_5)_{mod2} = 1)\}$ is not right-closed.

Theorem 3.2.7. *A PN structure N belongs to the class $\mathcal{H}$ if and only if $\forall p \in \Pi$, $\forall t_u \in p^\bullet \cap T_u$, we have:*

$$(\Gamma(p, t_u) = \min_{t \in p^\bullet} \Gamma(p, t)) \wedge (\forall t \in \Gamma(t_u), {}^\bullet t_u \subseteq {}^\bullet t)$$

Proof. (If) Suppose, $t \in T_e(N, \mathbf{m})$, for $\mathbf{m} \in \mathcal{N}^n$, and $\exists t_u \in \Omega(T) \cap T_u (\Rightarrow t \in \Omega(t_u))$. Since ${}^\bullet t_u \subseteq^\bullet t$ and $\forall p \in^\bullet, \Gamma(p, t_u) = min_{t \in p^\bullet} \Gamma(p, t)$, it follows that $t_u \in T_e(N, \mathbf{m})$.

(Only if) We will show that the violation of requirement in the statement if the theorem for a PN structure N would imply that $N \notin \mathcal{H}$. Suppose $\exists p \in \Pi, \exists t_u \in p^\bullet \cap T_u$ such that either

1. $\Gamma(p, t_u) > min_{t \in p^\bullet} \Gamma(p, t)$, or
2. $\exists t \in \Omega(t_u), ^\bullet t_u -^\bullet t \neq \varnothing$

In each of these cases we construct a marking $\mathbf{m} \in \mathcal{N}^n$, such that $\exists t \in \Omega(t_u) \cap T_e(N, \mathbf{m})$ and $t_u \notin T_e(N, \mathbf{m})$, which leads to the conclusion that $N \notin \mathcal{H}$.

For the first case, the marking $\mathbf{m}$ places exactly $(min_{t \in p^\bullet} \Gamma(p, t))$-many token in p, and sufficient tokens is the input places of any transition $\hat{t} \in \Omega(t_u)$ such that $\Gamma(p, \hat{t}) = min_{t \in p^\bullet} \Gamma(p, t)$ that will result in $\hat{t} \in T_e(N, \mathbf{m})$ and $t_u \notin T_e(N, \mathbf{m})$.

Similarly, for the second case, the marking $\mathbf{m}$ places sufficient tokens in the input places of t such that $t \in T_e(N, \mathbf{m})$, while ensuring that the places in $({}^\bullet t_u, ^\bullet t)$ remain empty. Consequently, $t \in T_e(N, \mathbf{m})$ and $t_u \notin T_e(N, \mathbf{m})$. □

There is an $O(n^2 m^2)$ algorithm that decides if an arbitrary PN structure belongs to the class $\mathcal{H}$, where $n = card(\Pi)$ and $m = card(T)$. The right-closure of $\Delta(N)$ for any $N \in \mathcal{H}$, along with the result in the reference [20], which is discussed in the previous section, implies that the existence of an LESP for $N(\mathbf{m}^0)$ is decidable. Furthermore, the software package described in reference [30] can be used to compute the minimally restrictive LESP for $N(\mathbf{m}^0)$ when it exists. Note that, each decidable class of PN structure identified in the references [20, 39, 40] are strictly contained in the class of $\mathcal{H}$. As an illustration, the PN structure N_5 shown in figure 3.1(b) is a member of $\mathcal{H}$ as it meets the structure requirement of theorem 3.2.7, and it does not belong to any of the classes of structure identified in the reference [20, 39, 40]. Additionally,

$$min(\Delta(N_5)) = \{(0\ 0\ 0\ 1\ 0)^T, (1\ 0\ 1\ 1\ 2)^T, (0\ 0\ 2\ 0\ 2)^T, (2\ 0\ 0\ 0\ 2)^T, (1\ 1\ 1\ 0\ 1)^T,$$

$$(0\ 1\ 2\ 0\ 1)^T, (2\ 1\ 0\ 0\ 1)^T, (0\ 2\ 2\ 0\ 0)^T, (1\ 2\ 1\ 0\ 0)^T, (2\ 2\ 0\ 0\ 0)^T\}$$

There is an LESP for $N_5(\mathbf{m}_5^0)$ if and only if $\mathbf{m}_5^0 \in \Delta(N_5)$.

A transition $t \in T$ is said to be a *choice-transition* (resp. *non-choice transition*) if $(^\bullet t)^\bullet \neq \{t\}$ (resp. $(^\bullet t)^\bullet = \{t\}$. In reference [41] it is shown that the minimally restrictive LESP for a *Free-choice* PNs does not control-disable a non-choice (controllable) transition. The following result shows that a similar observation holds for any minimally restrictive LESP for $N(\mathbf{m}^0)$ where $N \in \mathcal{H}$.

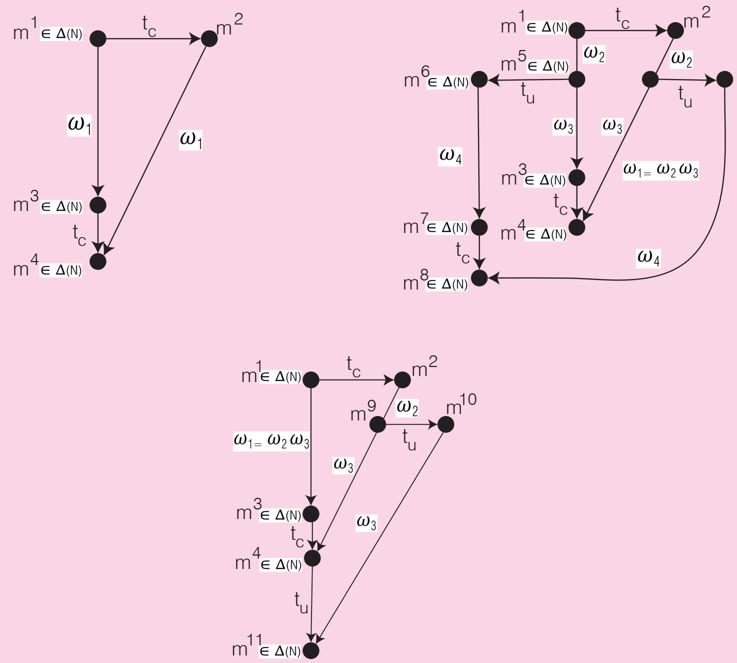


Figure 3.2: A graphical illustration used in the proof of theorem 3.2.8

Theorem 3.2.8. *[41] Suppose* $\mathbf{m}^0 \in \Delta(N)$ *for a PN* $N(\mathbf{m})^0$*, where* $N \in \mathcal{H}$*, then the minimally restrictive LESP* $\mathcal{P}^*$ *for* $N(\mathbf{m}^0)$ *does not disable any controllable transition* $t_c \in T_c$ *that satisfies the requirement* $(^\bullet t_c)^\bullet = \{t_c\}$

Proof. (Sketch) Suppose $\exists \mathbf{m}^1 \in \mathcal{R}(N, \mathbf{m}^0, \mathcal{P}^*)(\subseteq \Delta(N))$, $\exists \mathbf{m}^2 \in \mathcal{R}(N, \mathbf{m}^0)$, such that $\mathbf{m}^1 \overset{t_c}{\mathbf{m}}{}^2$ in N for some $t_c \in T_c$ that satisfies the requirement $(^\bullet t_c)^\bullet = \{T_c\}$. We will show that

1. $\exists \widetilde{\omega} \in T^*$ such that $\mathbf{m}^2 \overset{\widetilde{\omega}}{\to} \widetilde{\mathbf{m}}$ in N, where $\widetilde{m} \in \Delta(N)$.

2. Additionally, if $\widetilde{\omega} = \widetilde{\omega}_1\widetilde{\omega}_2$, $\mathbf{m}^2 \stackrel{\widetilde{\omega}_1}{\rightarrow} \widehat{\mathbf{m}}^1 \stackrel{\widetilde{\omega}_2}{\rightarrow} \widetilde{\mathbf{m}}$, and $\widehat{\mathbf{m}}^1 \stackrel{t_u}{\rightarrow} \widehat{\mathbf{m}}^2$ in N for some $t_u \in T_u$, then $\exists \widehat{\omega} \in T^*, \exists \widehat{\mathbf{m}}^3 \in \mathcal{R}(N, \widehat{\mathbf{m}}^2)$, such that $\widehat{\mathbf{m}}^1 \stackrel{t_u}{\rightarrow} \widehat{\mathbf{m}}^2 \stackrel{\widetilde{\omega}}{\rightarrow} \widehat{\mathbf{m}}^3$ and $\widehat{\mathbf{m}}^3 \in \Delta(N)$.

Following the repeated application of the above observation we conclude that $\mathbf{m}^2 \in \Delta(N)$

Since $\mathcal{P}^*$ is an LESP, $\exists \omega_1 \in (T - \{t_c\})^*$ and $\mathbf{m}^1 \stackrel{\omega_1}{\rightarrow} \widehat{\mathbf{m}}^2 \stackrel{t_c}{\rightarrow} \mathbf{m}^4$ in N under the supervision of $\mathcal{P}^*$. Since, $({}^\bullet t_c)^\bullet = \{t_c\}$, it follows that $\mathbf{m}^2 \stackrel{\omega_1}{\rightarrow} \mathbf{m}^4$ in N, and $\{\mathbf{m}^1, \mathbf{m}^3, \mathbf{m}^4\} \subseteq \Delta(N)$. Suppose $\omega_1 = \omega_2\omega_3$, $\mathbf{m}^2 \stackrel{\omega_2}{\rightarrow} \mathbf{m}^9 \stackrel{\omega_3}{\rightarrow} \mathbf{m}^4$, and $t_u \in T_u$ such that $\mathbf{m}^9 \stackrel{t_u}{\rightarrow} \mathbf{m}^{10}$. Also, $\exists \mathbf{m}^2 (\in \Delta(N))$ such that $\mathbf{m}^1 \stackrel{\omega_2}{\rightarrow} \mathbf{m}^5$. There are two cases to consider

1. $t_u \in T_e(N, \mathbf{m}^5)$ and,

2. $t_u \notin T_e(N, \mathbf{m}^5)$

In the first case, $\exists \mathbf{m}^6 (\in \Delta(N))$ such that $\mathbf{m}^5 \stackrel{t_u}{\rightarrow} \mathbf{m}^6$ (figure 3.2 (b)). Since $\mathcal{P}^*$ is an LESP, $\exists \omega_4 \in (T - \{t_c\})^*$, $\exists \mathbf{m}^7, \mathbf{m}^8 \in \Delta(N)$, such that $\mathbf{m}^6 \stackrel{\omega_4}{\rightarrow} \mathbf{m}^7 \stackrel{t_c}{\rightarrow} \mathbf{m}^8$. Since, $({}^\bullet t_c)^\bullet$, we have $\mathbf{m}^{10} \stackrel{\omega_4}{\rightarrow} \mathbf{m}^4$, where $\mathbf{m}^4 \in \Delta N$.

For the second case, where $t_u \notin T_e(N, \mathbf{m}^5)$, since $t_u \in T_e(N, \mathbf{m}^9)$, it follows that $\exists p \in \Pi$ such that $\{(t_c, p), (p, t_u)\} \subseteq \Phi$, and the prior-firing of t_c is necessary to place sufficient tokens in $p \in \Pi$, for t_u to be state-enabled at $\mathbf{m^5}$ (figure 3.2 (c)). Since $N \in \mathcal{H}$, it follows that none of the transitions in $\Omega(t_u)$ can fire at any marking that is reachable in the segment identified by $\mathbf{m}^1 \stackrel{\omega_1}{\rightarrow} \mathbf{m}^3$. Consequently, $t_u \in T_e(N, \mathbf{m}^4)$, and $\mathbf{m}^4 \stackrel{t_u}{\rightarrow} \mathbf{m}^{11}$ under the supervision of $\mathcal{P}^*$, where $\mathbf{m}^{11} \in \Delta(N)$. Consequently, $\mathbf{m}^{10} \stackrel{\omega_4}{\rightarrow} \mathbf{m}^{11}$ $\square$

Theorem 3.2.8 does not hold for general PN structure. The PN structure N_6 shown in figure 3.1(c) does not belong to the class $\mathcal{H}$. This is because $min_{t \in p_5^\bullet} \Gamma(p_5, t) = 1$, while $\Gamma(p_5, t_6) = 2$, and $t_6 \in p_5^\bullet \cap T_u$. Additionally, $\Delta(N_6) = \{\mathbf{m} \in \mathcal{N}^5 \mid (\mathbf{m}(p_1) + \mathbf{m}(p_2) + \mathbf{m}(p_3) + \mathbf{m}(p_4) \geq 1) \vee (\mathbf{m}(p_5)_{mod2} = 1)\}$, which is not right-closed. The minimally restrictive LESP for N_6, for any $\mathbf{m}_3^0 \in \Delta(N_6)$ control-disables a controllable transition at a marking in $\Delta(N_6)$ only if its firing result in a new marking that is not $\Delta(N_6)$. The minimally restrictive LESP would control-disable the non-choice transition $t_2 \in T_c$ at the marking $(0\ 1\ 0\ 0\ 1)^T \in \Delta(N_6)$.

As a consequence of theorem 3.2.8, without loss of generality, we can assume all non-choice transitions are uncontrollable, even when they are not, for any instance of the class of PN structures $\mathcal{H}$. This is formally stated in the following theorem.

Theorem 3.2.9. *Let $N = (\Pi, T, \Phi, \Gamma)$ be a PN structure from the family $\mathcal{H}$, where the set of transition is partitioned into the set of uncontrollable transitions T_u, and controllable transitions T_c (i.e $T = T_c \cup T_u$ and $T_u \cap T_c = \varnothing$). Suppose $\widehat{N}$ is another member of the family $\mathcal{H}$ that is structurally identical to N, but the set of transitions in $\widehat{N}$ are partitioned into a different set of uncontrollable and controllable transitions, where*

$$\widehat{T}_u = T_u \cup \{t \in T \mid ({}^\bullet t)^\bullet = \{t\}\}$$

and $\widehat{T}_c = T - \widehat{T}_u$. Then $\Delta(N) = \Delta(\widehat{N})$.

Proof. Since $\widehat{T}_c \subseteq Tc$, it follows that $\Delta(\widehat{N}) \subseteq \Delta(N)$. The reverse inclusion is shown by contradiction. Suppose, $\Delta(\widehat{N}) \subset \Delta(N)$, then $\Delta(N)$ is not control invariant with respect to $\widehat{N}$. That is $\exists \mathbf{m}^1 \in \Delta(N), \exists \hat{t}_u \in \widehat{T}_u$, such that $\mathbf{m}^1 \xrightarrow{t_u} \mathbf{m}^2$ and $\mathbf{m}^2 \notin \Delta(N)$. Since, $\Delta(N)$ is control invariant with respect to N, it must be that $\hat{t}_u \in \{t \in T \mid ({}^\bullet t)^\bullet = \{t\}\}$. But, from theorem 3.2.8, we know that $\mathbf{m}^2 \in \Delta(N)$, which establishes the result. □

As an illustration, the non-choice, controllable transition t_2 in the PN structure N_5 of figure 3.1(b) can be considered to be uncontrollable, which effectively results in a PN structure with no controllable transitions. There is an LESP for a PN $N(\mathbf{m}^0)$ with no controllable transitions if and only if the PN is live. This leads to the observation that the PN $N_5(\mathbf{m}_5^0)$ is live for any $\mathbf{m}_5^0 \in \Delta(N_5)$.

The observation that we can assume all non-choice transitions are uncontrollable, even when they are not for any $N \in \mathcal{H}$, is critical to the speeding-up the execution of the software package described in reference [30].

3.2.3 Why Convexity?

The existence of a monitor-placement does not guarantee its equivalence to a minimally restrictive LESP. This section discusses results from reference [25], which is about the necessary and sufficient condition under which an invariant-based monitor exists which is equivalent to the LESP $\widetilde{\mathcal{P}}$ that ensures the set of reachable markings under its supervision stays within some subset $\widetilde{\Delta}(N) \subseteq \Delta(N)$. Note that, we assume that the PNs has an computable minimally restrictive LESP, which here, it translates to having a right-closed $\Delta(N)$ or a right-closed subset of $\Delta(N)$. For following theorem, we use lemma 4.1.2, which will be introduced in the next chapter for readability.

Theorem 3.2.10. *Let* $\widetilde{\Delta}(N) \subseteq \Delta(N)$ *be a right-closed set that is control-invariant with respect to a PN structure* $N = (\Pi, T, \Phi, \Gamma)$*. Further, let us suppose that each member of* $\min(\widetilde{\Delta}(N))$ *meets the path-requirement mentioned earlier. There exists an invariant-based monitor that is equivalent to the LESP* $\widetilde{\mathcal{P}}$ *that ensures* $\Re(N, \mathbf{m}^0, \widetilde{\mathcal{P}}) \subseteq \widetilde{\Delta}(N)$ *for each* $\mathbf{m}^0 \in \widetilde{\Delta}(N)$*, if and only if* $\widetilde{\Delta}(N)$ *is convex.*

Proof. (If) If the right-closed set $\widetilde{\Delta}(N)$ is convex, then from lemma 4.1.2, it can be represented as, $Int(P(\mathbf{A}, \mathbf{b}))$, the set of integral points in a polyhedron $P(\mathbf{A}, \mathbf{b})$ in the positive orthant. From the results in references [28, 29], the invariant-based monitor $N_c(\mathbf{m}^0, \Theta(\mathbf{m}^0))$, where $\Theta(\mathbf{m}^0) = \mathbf{A}\mathbf{m}^0 - \mathbf{b}$, and an incidence matrix for the monitors that is given by $\mathbf{AC}$, will guarantee $\forall(\mathbf{m}_1^T \ \mathbf{m}_2^T)^T \in \Re(N, \mathbf{m}^0, \Theta(\mathbf{m}^0)), \mathbf{A}\mathbf{m}_1 \geq \mathbf{b}$, and $\mathbf{m}_2^T = \mathbf{A}\mathbf{m}_1 - \mathbf{b} \geq \mathbf{0}$.

The control-invariance of $\widetilde{\Delta}(N)$ with respect to N guarantees that any firing of a state-enabled uncontrollable transition at some marking in $\widetilde{\Delta}(N)$ in N will always result in a marking that is in $\widetilde{\Delta}(N)$. Consequently, in $N_c(\mathbf{m}^0, \Theta(\mathbf{m}^0))$, if there are any arcs from a monitor to an uncontrollable transition, it will not be prevented from firing due to insufficient tokens in the monitor. On the flip-side, if there is a controllable transition with a monitor as one of its input places, that is prevented from firing only because there are insufficient tokens in a monitor at some marking, then it must be that the new marking that would result if the controllable transition were permitted to fire would violate the (invariance) requirement $\mathbf{A}\mathbf{m}_1 \geq \mathbf{b}$, and $\mathbf{m}_1 \notin \widetilde{\Delta}(N)$. Therefore, the control exerted by the monitors in $N_c(\mathbf{m}^0, \Theta(\mathbf{m}^0))$ is the same as the LESP $\widetilde{\mathcal{P}}$.

(Only If) Suppose for any $\mathbf{m}^0$ such that $\mathbf{A}\mathbf{m}^0 \geq \mathbf{b}$, there is an invariant-based monitor $N_c(\mathbf{m}^0, \Theta(\mathbf{m}^0))$, where $\Theta(\mathbf{m}^0) = \mathbf{A}\mathbf{m}^0 - \mathbf{b}$, and an incidence matrix for the monitors that is given by $\mathbf{AC}$, that is equivalent to the LESP $\widetilde{\mathcal{P}}$ that ensures $\Re(N, \mathbf{m}^0, \widetilde{\mathcal{P}}) \subseteq \widetilde{\Delta}(N)$. Then, $\widetilde{\Delta}(N) = Int(P(\mathbf{A}, \mathbf{b}))$, which in turn implies $\widetilde{\Delta}(N)$ is convex. □

Therefore, in order to answer the question of existence of an invariant-based monitor that is equivalent to the LESP, we need to first answer the question of convexity of the underlying right-closed set. Earlier, in figure 2.2(a) we introduced a PN $N_3(\mathbf{m}_3^0)$ and its minimally restrictive LESP that used forty-three vectors, identified in figure 2.2(b). It is worthwhile to note that the PN structure N_3 is an ordinary FCPN, which would mean that $\Delta(N_3)$ is right-closed. The forty-three vectors of figure 2.2(b) are essentially the members of $\min(\Delta(N_3))$ that were computed using the software of reference [30]. It can be shown that $\Delta(N_3)$ is convex, which in turn means there is monitor construction (cf. figure 2.3) that is equivalent to the minimally restrictive LESP of 2.2(a).

Chapter 4

Convexity of Right-Closed Set

A set in euclidean space is called convex if for every pair of points in the set, the line segment connecting them, lies completely in the set. While a well-established concept, unfortunately checking this property for an arbitrary geometric set is not easy. For example, consider deleting a single point from a convex set; determining that the set is non-convex is almost impossible as infinite pairs of points should be tested [21]. Verifying convexity of a set of integral vectors gets even worse, as all the tests for convexity over the real sets will collapse when dealing whit an integral set. In this chapter, we present results which show that testing convexity for a right-closed integral set is decidable.

4.1 Polyhedral Representation

We start this section with a simple but important lemma about a polyhedral representation of an arbitrary right-closed set of integral points. Since we concern ourselves with polyhedra that are in the positive orthant, we implicitly require $\forall \mathbf{x} \in P(\mathbf{A}, \mathbf{b}), \mathbf{x} \geq \mathbf{0}$ in this section.

Lemma 4.1.1. *For $i, j \in \mathcal{N}$, $\mathbf{A} \in \mathcal{R}^{i \times j}$, $\mathbf{b} \in \mathcal{R}^i$, the set of integral points, $Int(P(\mathbf{A}, \mathbf{b})) = P(\mathbf{A}, \mathbf{b}) \cap \mathcal{N}^n$, in a polyhedron $P(\mathbf{A}, \mathbf{b})$ is right-closed if and only if $\mathbf{A}$ is non-negative.*

Proof. (*Only If*) Suppose $Int(P(\mathbf{A}, \mathbf{b}))$ is right-closed and $\mathbf{A}_{l,m}$, the (l, m)-th entry of $\mathbf{A}$, is negative. From the right-closure of $Int(P(\mathbf{A}, \mathbf{b}))$, we have $(\mathbf{x} \in Int(P(\mathbf{A}, \mathbf{b}))) \wedge (\widehat{\mathbf{x}} \geq \mathbf{x}) \Rightarrow (\widehat{\mathbf{x}} \in Int(P(\mathbf{A}, \mathbf{b})))$. Suppose, $\widehat{\mathbf{x}}_k = \mathbf{x}_k, \forall k \in \{1, 2, \ldots, j\} - \{m\}$, and if $\widehat{\mathbf{x}}_m$ is made arbitrarily large compared to $\mathbf{x}_m$, the l-th component of $\mathbf{A}\widehat{\mathbf{x}}$ can be made less than $\mathbf{b}_m$. This would mean $\widehat{\mathbf{m}} \notin Int(P(\mathbf{A}, \mathbf{b}))$. A contradiction.

(*If*) Follows directly from the definition of $Int(P(\mathbf{A}, \mathbf{b}))$. □

Additionally, without loss of generality $\mathbf{b}$, can be assumed to be non-negative if $P(\mathbf{A}, \mathbf{b})$ is in the positive orthant, as with a polyhedron whose integral points define a set of markings of a PN. If $\mathbf{A}$ is a non-negative matrix and $\mathbf{b}_l < 0$, then $\mathbf{b}_l \leftarrow 0$ will yield the same polyhedron as before. Consequently, $\mathbf{b}$ can be assumed to be non-negative as well for these instances.

Lemma 4.1.2. *A right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$ is convex if and only if $\mathcal{M} = Int(P(\mathbf{A}, \mathbf{b}))$, for a polyhedron for some non-negative $\mathbf{A}$ and a non-negative $\mathbf{b}$.*

Proof. (If) If $\mathbf{A}$ and $\mathbf{b}$ are non-negative, then $Int(P(\mathbf{A}, \mathbf{b}))$ is a (convex) right-closed set (cf. Lemma 4.1.1). Consequently, $\mathcal{M} = Int(P(\mathbf{A}, \mathbf{b}))$ is a convex, right-closed set as well.

(Only If) A convex right-closed set $\mathcal{M}$ can be written as the set of integral points in a set $\mathcal{A}$ that is the Minkowski sum of the convex combination of the members of $\min(\mathcal{M})$ and the cone generated by the unit-vectors. From the *Affine Minkowski-Weyl Duality Theorem* we infer that $\mathcal{A}$ is a polyhedron. From lemma 4.1.1 and the discussion that followed it, we can assume the polyhedron is of the form $P(\mathbf{A}, \mathbf{b})$, where $\mathbf{A}$ and $\mathbf{b}$ are non-negative. □

Conversely, if $\mathcal{M} \subseteq \mathcal{N}^n$ is a right-closed set that is not convex, there can be no polyhedron $P(\mathbf{A}, \mathbf{b})$ such that $\mathcal{M} = Int(P(\mathbf{A}, \mathbf{b}))$. Or, any right-closed, polyhedral approximation to a non-convex, right-closed $\mathcal{M} \subseteq \mathcal{N}^n$ will inevitably exclude some members.

4.2 Convexity Condition

Following the results for polyhedral presentation of a right-closed set, we present a necessary and sufficient condition under which, an arbitrary right-closed integral set is convex.

Lemma 4.2.1. *A set $\mathcal{M} \subseteq \mathcal{N}^n$ is convex if and only if $\mathcal{M} = Int(conv(\mathcal{M}))$.*

Proof. (Only if) Since $\mathcal{M}$ is convex, there exists a convex set $\mathbf{C} \subseteq \mathcal{R}^n$ such that $\mathcal{M} = Int(\mathbf{C})$. By definition, $conv(\mathcal{M})$ is the smallest convex set that contains all members of $\mathcal{M}$. Therefore, $conv(\mathcal{M}) \subseteq \mathbf{C}$ and $conv(\mathcal{M}) \cap \mathcal{N}^n = \mathbf{C} \cap \mathcal{N}^n (= \mathcal{M})$.

(If) Follows directly from the definition of convexity. □

Lemma 4.2.2 notes that the set of integral vectors in the convex-hull of a right-closed set is also right-closed.

Lemma 4.2.2. *The set $Int(conv(\mathcal{M}))$ is right-closed for any $\mathcal{M} \subseteq \mathcal{N}^n$ that is right-closed.*

Proof. Suppose $\widehat{\mathbf{m}} \in Int(conv(\mathcal{M}))$, then without loss of generality $\widehat{\mathbf{m}} = \sum_{i=1}^{k} \lambda_i \mathbf{m}_i$, where $\forall i \in \{1, 2, \ldots, k\}, \mathbf{m}_i \in \mathcal{M}$ and $\sum_{i=1}^{k} \lambda_i = 1$. If $\mathbf{1}_j \in \mathcal{N}^n$ is any one of the n-many unit vectors of $\mathcal{N}^n$, then $(\widehat{\mathbf{m}} + \mathbf{1}_j) = \sum_{i=1}^{k} \lambda_i (\mathbf{m}_i + \mathbf{1}_j)$, and $(\mathbf{m}_i + \mathbf{1}_j) \in \mathcal{M}$ as $\mathcal{M}$ is right-closed. Therefore, $(\widehat{\mathbf{m}} + \mathbf{1}_j) \in conv(\mathcal{M})$. The result follows from this observation. □

As a consequence of lemma 4.2.2, it follows that, $\min(Int(conv(\mathcal{M})))$, the set of minimal elements of $Int(conv(\mathcal{M}))$ is finite. Theorem 4.2.3 yields an effectively computable test for the convexity of a right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$ and is a direct consequence of Lemmas 4.2.1 and 4.2.2.

Theorem 4.2.3. *A right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$ is convex if and only if $\min(\mathcal{M}) = \min(Int(conv(\mathcal{M})))$. Equivalently, $\mathcal{M} \subseteq \mathcal{N}^n$ is convex if and only if $\mathcal{M} = Int(conv(\mathcal{M}))$.*

4.2.1 Some Additional Observations and an Illustrative Example

Although aforementioned theorems will be a strong base to test the convexity of a right-closed set, but it is also important to develop an effectively computable test for this test. Computing the $\min(Int(conv(\mathcal{M})))$ can be a very computationally expensive task. Following is a very helpful theorem in computing the desired set. An illustrative example will show the importance of this theorem.

The procedure to compute the convex hull of a set of vectors is foundational to this section. The following result shows that this can be done in polynomial time.

Theorem 4.2.4. *[35] It is possible to compute the convex hull of m points in n-space deterministically in $\mathcal{O}(mlog(m) + m^{\lfloor n/2 \rfloor})$*

The following lemma identifies a property of the convex hull of $\min(\mathcal{M})$ where $\mathcal{M} \subseteq \mathcal{N}^n$ is an arbitrary right-closed set.

Lemma 4.2.5. *If $\mathcal{M} \subseteq \mathcal{N}^n$ is an arbitrary right-closed set, then all vertices of the convex hull of* $\min(\mathcal{M})$ *are minimal elements.*

Proof. This observation follows directly from the fact that every vector in $conv(\min(\mathcal{M}))$ is a convex combination of the finite set of vectors in $min(\mathcal{M})$. □

As the theorem 4.2.4 states, computing the convex hull of finite points in a finite dimension has a polynomial-time complexity with respect to number of points for a fixed dimension. We will utilize this theorem as a basis for our proposed algorithm presented in the next chapter. But in order to construct an efficient procedure for testing the convexity, we cannot directly compute the convex hull of the set $\mathcal{M}$, as this set is an infinite set by definition. To overcome this issue, we proposed a finite subset of the original right-closed set, $\mathcal{V}$, which keeps the characteristics that can attest to the convexity of the original right-closed set.

Let $\mathcal{V} \subset \mathcal{M}$ be a finite collection of vectors as defined below:

$$\mathcal{V} = min(\mathcal{M}) \cup \left\{ \bigcup_{p \in \{1,\cdots,k\}, q \in \{1,\cdots,n\}} \{\mathbf{m}_p + \mathbf{1}_q\} \right\} \tag{4.1}$$

where $\min(\mathcal{M}) = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_k\}$ and $\{\mathbf{1}_q\}_{q=1}^{n}$ is the set of n-many unit-vectors. In general, $\min(\mathcal{M}) \subseteq \min(Int(conv(\mathcal{V})))$. The following result notes that for a non-convex right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$, $\min(Int(conv(\mathcal{V}))) - \mathcal{M} \neq \emptyset$.

The finite number of integral vectors in the polytope $conv(\mathcal{V})$ can be enumerated using software packages like *Polymake* [33, 34], which can be subsequently processed to compute the members of $min(Int(conv(\mathcal{M})))$. Consequently, we have a procedure for testing the convexity of $\mathcal{M}$ 4.2.3.

Lemma 4.2.6. *Let $min(\mathcal{M}) = \{m_1, m_2, \cdots, m_k\}$ be the minimal elements of a right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$. Then $min(Int(conv(\mathcal{M}))) \subseteq conv(\mathcal{V})$, where $\mathcal{V} = min(\mathcal{M}) \cup \{\bigcup_{p \in \{1,\cdots,k\}, q \in \{1,\cdots,n\}} \{\mathbf{m}_p + 1_q\}\}$.*

Proof. Let $x \in min(Int(conv(\mathcal{M})))$, and $x = (\sum_{i \in I} \lambda_i m_i) + (\sum_{j \in J} \mu_j p_j)$ for appropriate index sets I and J, where $\{p_j\}_{j \in J}$ is a set of integral vectors in the set $(\mathcal{M} - min(\mathcal{M}))$, $\sum_{i \in I} \lambda_i + \sum_{j \in J} \mu_j = 1$, $\forall i \in I, \lambda_i \geq 0$, and $\forall j \in J, \mu_j \geq 0$. Let us suppose each $p_j > \widehat{m}_j$ for $j \in J$, and $\widehat{m}_j \in min(\mathcal{M})$. Then

$x = (\sum_{i \in I} \lambda_i m_i) + (\sum_{j \in J} \mu_j p_j) \geq x = (\sum_{i \in I} \lambda_i m_i) + (\sum_{j \in J} \mu_j \widehat{m}_j) = \sum_{i \in K} \lambda_i m_i = y$, where the set K is obtained from the index set I and J through the appropriate operations.

Since $y \leq x$, and $x \in min(Int(conv(\mathcal{M})))$, we have $x = \lceil y \rceil$. For any pair of sets $\mathcal{A}$ and $\mathcal{B}$, $conv(\mathcal{A}+\mathcal{B}) = conv(\mathcal{A}) + conv(\mathcal{B})$, where the summation operator is the Minkowski sum. This, together with the fact that $\lceil y \rceil = y + w$, where $w \in conv(\{0\} \cup \{1_q\}_{q=1}^n)$ and $y \in conv(min(\mathcal{M}))$, it follows that $x \in conv\mathcal{V}$ □

Therefore, we can simply infer that:

Theorem 4.2.7. *A right-closed set* $\mathcal{M} \subseteq \mathcal{N}^n$ *is convex if and only if* $\min(\mathcal{M}) = \min(Int(conv(\mathcal{V})))$.

4.3 Computing $\min(Int(conv(\mathcal{M})))$: Illustrative Examples

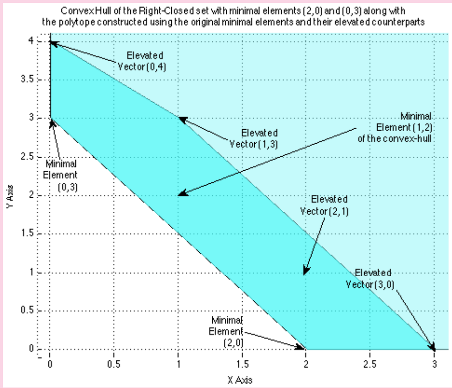
We turn our attention to a procedure for computing $min(Int(conv(\mathcal{M})))$, which is introduced via examples in this section.

Consider a right-closed set $\mathcal{M}_1$, where $\min(\mathcal{M}_1) = \{(2\ 0)^T, (0\ 3)^T\}$. Figure 4.1(a) shows a truncated view of $conv(\mathcal{M}_1)$ along with the $\mathcal{V}$-polytope for this example, defined by the vertex set $\{(2\ 0)^T$, $(3\ 0)^T$, $(2\ 1)^T$, $(0\ 3)^T$, $(1\ 3)^T$, $(1\ 4)^T\}$. As noted in the caption of figure 4.1(a), we conclude $\mathcal{M}_1$ is not convex. The facet defining hyperplanes for $conv(\mathcal{M}_1)$ are $F1 : x \geq 0, F2 : -x - y \geq -4, F3 : 3x + 2y \geq 6, F4 : y \geq 0$, and $F5 : -3x - 2y \geq -9$. Of these, $F1$ and $F4$ are ignored as the convex hull is (trivially) in the positive orthant. Of these, only $F3$ defines a right-closed set and consequently $conv(\mathcal{M}_1)$ is identified by the inequality $(3\ 2)\mathbf{x} \geq 6$ for $\mathbf{x} \in \mathcal{R}^2$ $(\mathbf{x} \geq \mathbf{0})$.

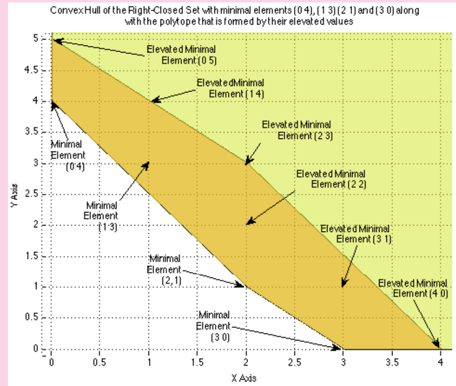
We now consider $\mathcal{M}_2 \subset \mathcal{M}_1$ identified by minimal elements

$$\{(0\ 4)^T, (1\ 3)^T, (2, 1)^T, (3\ 0)^T\}.$$

Figure 4.1(b) shows a truncated version of $conv(\mathcal{M}_2)$ along with the (V)-polytope defined by the vertex set that corresponds to these minimal elements together with their elevated counterparts. As noted in the caption of figure 4.1(b), we conclude that $\mathcal{M}_2$ is convex. The facet defining hyperplanes for



(a) $conv(\mathcal{M}_1)$



(b) $conv(\mathcal{M}_2)$

Figure 4.1: The convex hull of $\mathcal{M}_1$, where $\min(\mathcal{M}_1) = \{(2\ 0)^T, (0\ 3)^T\}$, along with the $\mathcal{V}$-polytope that is defined by the vertex set $\{(2\ 0)^T, (3\ 0)^T, (2\ 1)^T, (0\ 3)^T, (1\ 3)^T, (1\ 4)^T\}$. The set of minimal elements of the seven integral vectors in this polytope, $\min(Int(conv(\mathcal{M}_1)))$, is $\{(0\ 3)^T, (1\ 2)^T, (2\ 0)^T\} (\neq \min(\mathcal{M}_1))$, therefore $\mathcal{M}_1$ is not convex. (b) The convex hull of $\mathcal{M}_2$, where $\min(\mathcal{M}_2) = \{(0\ 4)^T, (1\ 3)^T, (2\ 1)^T, (3\ 0)^T\}$. The $\mathcal{V}$-polytope defined by these minimal elements and their elevated counterparts has ten integral vectors, and $\min(Int(conv(\mathcal{M}_2))) = \{(0\ 4)^T, (1\ 3)^T, (2\ 1)^T, (3\ 0)^T\} (= \min(\mathcal{M}_2))$, which implies that $\mathcal{M}_2$ is convex.

$conv(\mathcal{M}_2)$ are $(1\ 1)\mathbf{x} \geq 3$ and $(3\ 2)\mathbf{x} \geq 8$ for $\mathbf{x} \in \mathcal{R}^2$ $(\mathbf{x} \geq \mathbf{0})$. However, if $\mathbf{x} \in \mathcal{N}^2$, $((3\ 2)\mathbf{x} \geq 8) \Rightarrow ((1\ 1)\mathbf{x} \geq 3)$, and consequently $\mathcal{M}_2 = \{\mathbf{x} \in \mathcal{R}^2 \mid (3\ 2)\mathbf{x} \geq 8\} \cap \mathcal{N}^2$.

For the final example, consider the right-closed set $\mathcal{M}_3 \subseteq \mathcal{N}^3$ that is defined by the minimal elements $\{(1\ 4\ 2)^T, (3\ 5\ 1)^T, (5\ 1\ 3)^T\}$. Figures 4.2(a) and 4.2(b) shows two different, truncated views of the unbounded convex-hull of $\mathcal{M}_3$. At noted earlier, it is of interest to compute the minimal integral vectors in $conv(\mathcal{M}_3)$, and to this end we introduced the $\mathcal{V}$-polytope, that (1) shares (right-closed) facets with $conv(\mathcal{M}_3)$, and (2) contains all members of $min(Int(conv(\mathcal{M}_3)))$. Figures 4.2(c) and 4.2(d) shows two different views of such a polytope, along with its position relative to $conv(\mathcal{M})$. This polytope is defined by a vertex set consisting of each member of $min(\mathcal{M}_3)$, along with three elevated versions of each member of $min(\mathcal{M}_3)$ along each of the three axes. That is, the polytope is defined by the vertex set $\{(1\ 4\ 2)^T, (2\ 4\ 2)^T, (1\ 5\ 2)^T, (1\ 4\ 3)^T, (3\ 5\ 1)^T, (4\ 5\ 1)^T, (3\ 6\ 1)^T, (3\ 5\ 2)^T, (5\ 1\ 3)^T, (6\ 1\ 3)^T, (5\ 2\ 3)^T, (5\ 1\ 4)^T\}$. Members of $min(Int(conv(\mathcal{M}_3)))$ are the minimal elements of the integral vectors in this newly introduced polytope, Using *Polymake*, we note there are twenty integral points in this polytope, which includes the six minimal elements – $\{(1\ 4\ 2)^T, (3\ 3\ 3)^T, (3\ 5\ 1)^T, (4\ 2\ 3)^T, (4\ 3\ 2)^T, (5\ 1\ 3)^T\}$. Since $min(Int(conv(\mathcal{M}_3))) \neq min(\mathcal{M})$, from theorem 4.2.3 it follows that $\mathcal{M}_3$ is not convex. This is verified by noting

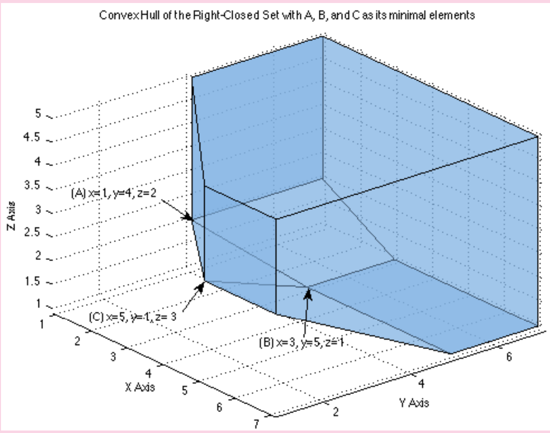
$$\underbrace{(3\ 3\ 3)^T}_{\notin \mathcal{M}} = \frac{4}{10}\underbrace{(1\ 4\ 4)^T}_{\in \mathcal{M}} + \frac{2}{10}\underbrace{(3\ 5\ 1)^T}_{\in \mathcal{M}} + \frac{4}{10}\underbrace{(5\ 1\ 3)^T}_{\in \mathcal{M}}.$$

From lemmas 4.1.2 and 4.2.2 we infer that $conv(\mathcal{M}_3) = P(\mathbf{A}, \mathbf{b})$ for an appropriately defined, non-negative $\mathbf{A}$ and $\mathbf{b}$. The newly introduced polytope can be used to identify an appropriate $\mathbf{A}$ and $\mathbf{b}$ that defines $conv(\mathcal{M}_3)$. *Polymake* identifies ten facet defining hyperplanes in the polytope: $F1 : x+3y+5z \geq 23$, $F2 : -y-z \geq -7$, $F3 : x+2z \geq 5$, $F4 : x \geq 1$, $F5 : z \geq 1$, $F6 : -x-3y-5z \geq -28$, $F7 : x+\frac{4}{3}y \geq \frac{19}{3}$, $F8 : y \geq 1$, $F9 : y+2z \geq 7$, and $F10 : -x-y-z \geq -10$. Of these, $F1, F3, F4, F5, F7, F8$ and $F9$ define right-closed sets, and their intersection essentially defines $conv(\mathcal{M}_3)$. Therefore, $conv(\mathcal{M}_3) = P(\mathbf{A}, \mathbf{b})$, where

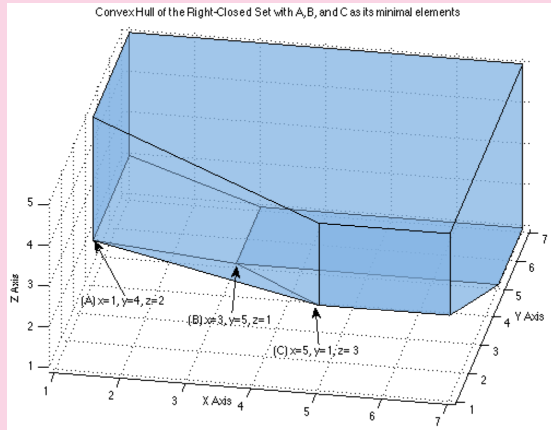
$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 5 \\ 1 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 3 & 4 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} 23 \\ 5 \\ 1 \\ 1 \\ 19 \\ 1 \\ 7 \end{pmatrix}$$

4.4 Convexity Testing

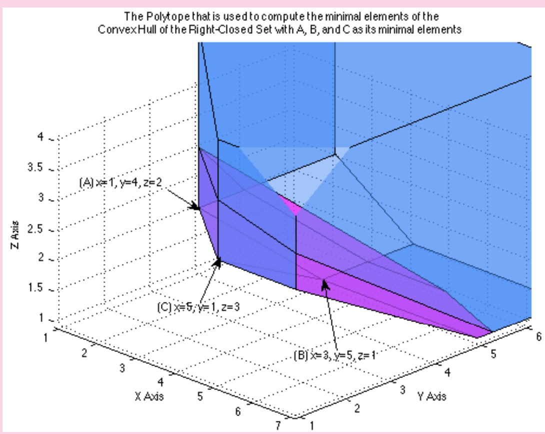
Unfortunately, the proposed test for checking the convexity, still is not efficient. The volume of $conv(\mathcal{V})$ can increase drastically if the dimension grows. Distribution of the minimal elements in the space can also contribute to it. This increase in volume, will also effect the computation time of deriving the minimal elements of $conv(\mathcal{V})$ adversely. Hence, eliminating the procedure of computing the minimal elements for testing the convexity is desired. In the following section, we look deeper into properties



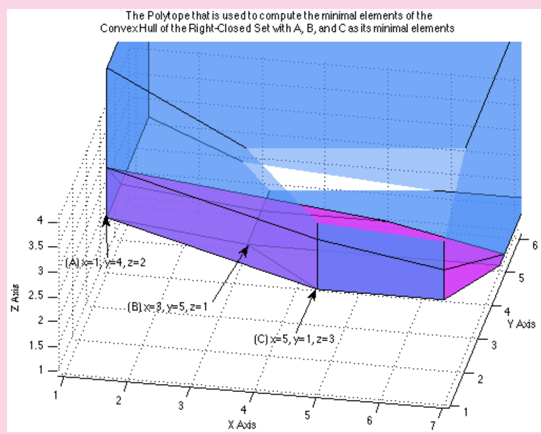
(a) Convex Hull - View 1



(b) Convex Hull - View 2



(c) Convex Hull & Polytope - View 1



(d) Convex Hull & Polytope - View 2

Figure 4.2: Figures (a) and (b) present two different views of (the polyhedral representation of) $conv(\mathcal{M}_3)$. The minimal elements of $\mathcal{M}_3$, represented using coordinates $(x\ y\ z)^T$, are $A = (1\ 4\ 2)^T, B = (3\ 5\ 1)^T$ and $C = (5\ 1\ 3)^T$. Although the convex-hull is unbounded, the above plot is limited to the ranges $1 \leq x \leq 7$, $1 \leq y \leq 7$ and $1 \leq z \leq 5$ for illustration purposes. The minimal elements of $Int(conv(\mathcal{M}))$ are restricted to lie within a polytope, shown in figures (c) and (d), that is defined by a vertex set that is formed by each member of $\min(\mathcal{M}_3)$, that is subsequently elevated by three unit-vectors. That is, the polytope is defined by the vertex set $\{(1\ 4\ 2)^T, (2\ 4\ 2)^T, (1\ 5\ 2)^T, (1\ 4\ 3)^T, (3\ 5\ 1)^T, (4\ 5\ 1)^T, (3\ 6\ 1)^T, (3\ 5\ 2)^T, (5\ 1\ 3)^T, (6\ 1\ 3)^T, (5\ 2\ 3)^T, (5\ 1\ 4)^T\}$. The minimal elements of the integral points of this polytope are the minimal elements of $Int(conv(\mathcal{M}_3))$.

of the set $conv(\mathcal{V})$ in order to find a better way to tackle this problem. We show that constructing a subset of $conv(\mathcal{M})$ in a slightly different way, while keeping the properties of $conv(\mathcal{V})$, results in efficient procedures for testing the convexity.

The finite set $\mathcal{V}$ is a strict subset of $\mathcal{M}$, hence we can infer that the set of supporting hyperplanes defining $conv(\mathcal{M})$ is a strict subset of supporting hyperplanes defining $conv(\mathcal{V})$. On the other hand, $conv(\mathcal{M})$ is a right-closed set which means all its supporting hyperplanes are right-closed hyperplanes. The following theorem shows that if a supporting hyperplane of $conv(\mathcal{V})$ is not a supporting hyperplane of $conv(\mathcal{M})$, then it is not right-closed.

Theorem 4.4.1. *The set of right-closed supporting hyperplanes defining $conv(\mathcal{V})$ are the only supporting hyperplanes defining $conv(\mathcal{M})$.*

Proof. Since $\mathcal{M} \subseteq \mathcal{N}^n$ is right-closed, $conv(\mathcal{M}) \subseteq \mathcal{R}^n$ is also right-closed. From reference [25], $\exists l \in \mathcal{N}, \exists \mathbf{A} \in \mathcal{R}^{l\times n}, \exists \mathbf{b} \in \mathcal{R}^l$, such that $conv(\mathcal{M}) = P(\mathbf{A}, \mathbf{b})$. As $conv(\mathcal{V}) \subset conv(\mathcal{M})$, extra constraints should be added to the mentioned polyhedron in order to construct $conv(\mathcal{V})$. Let $\mathbf{c}_i^T\mathbf{x} \geq d_i$ for $i = 1, \cdots, K$, be the K additional constraints. For any vector $\mathbf{p} \in conv(\mathcal{M}) - conv(\mathcal{V})$, we obtain

$$\mathbf{c}_i^T\mathbf{p} < d_i, \forall i = 1, \cdots, K$$

Additionally, for any $\mathbf{y} \geq 0$, $\mathbf{p} + \mathbf{y}$ also does not belong to $conv(\mathcal{V})$ either. Therefore,

$$\mathbf{c}_i^T\mathbf{p} + \mathbf{c}_i^T\mathbf{y} < d_i, \forall i = 1, \cdots, K.$$

Comparing these two expressions, suggests that $\mathbf{c}_i$ for $i = 1, \cdots, K$ has to have at least one negative component, regardless of the sign of d_i. Hence, the additional constraints in the form of $\mathbf{c}_i^T\mathbf{x} \geq d_i$ cannot be right-closed hyperplanes □

For $1 \leq i \leq n$, let γ_i denote the maximum value of the i-th component of each member of $\mathcal{V}$. That is, $\gamma_i := \max\{\mathbf{x}_i \mid \mathbf{x} \in \mathcal{V})\}$. We define $\widetilde{P} \subseteq \mathcal{R}^n$ as the polytope that is defined as:

$$conv(\mathcal{M}) \cap \{(-\mathbf{1}_i, -\gamma_i)\}_{i=1}^{n} \tag{4.2}$$

That is, each of the left-closed half-spaces ensures the i-th component is less than or equal to γ_i. Therefore $\widetilde{P}$ consists of either right-closed or left-closed hyperplanes.

Theorem 4.4.2. *The set of right-closed supporting hyperplanes defining $conv(\mathcal{V})$ are the only right-closed supporting hyperplanes defining $conv(\mathcal{M})$ and $\widetilde{P}$*

Proof. We first show that $conv(\mathcal{V}) \subset \widetilde{P}$. Notice that $conv(\mathcal{V}) \subset conv(\mathcal{M})$. Also all the points in $conv(\mathcal{V})$ satisfy the previously mentioned left-closed hyperplanes that are added to $conv(\mathcal{V})$ in order to construct the $\widetilde{P}$. Hence, every point in $conv(\mathcal{V})$ also belongs to $\widetilde{P}$.

Now we show that $conv(\mathcal{V})$ is constructed from $\widetilde{P}$ by adding only non-right-closed hyperplanes. Let $cx \geq d$ be the constraint added to $\widetilde{P}$ in order to construct the $conv(\mathcal{V})$. For every $v \in conv(\mathcal{V})$ we have $cv \geq d$. Define $\gamma_{max} = (\gamma_1, \cdots, \gamma_n)$. If c and d are to be non-negative, therefore $c\gamma_{max} \geq d$ as $\gamma_{max} \geq v$. But $\gamma_{max} \notin conv(\mathcal{V})$ by its definition although $\gamma_{max} \in \widetilde{P}$. Therefore $c\gamma_{max} < d$. This contradicts the assumption that both c, d are non-negative. Hence, the additional constraints to $\widetilde{P}$ cannot be right-closed. □

Therefore, the (half-space) description of $\widetilde{P}$ can be obtained by adding additional half-spaces in the form of $\{(-1, \gamma_i)\}_{i=1}^{n}$ to the set of right-closed facets of $\mathcal{V}$. This presents an computational procedure for constructing the polytope $\widetilde{P}$. It is straightforward to show that $\min(Int(conv(\mathcal{M}))) = \min(Int(\widetilde{P}))$.

Theorem 4.4.3. *$\forall \boldsymbol{x} \in min(Int(\widetilde{P}))$, there is at least one right-closed facet of $\widetilde{P}$, $\mathcal{F}$, such that $\alpha \boldsymbol{x} \in \mathcal{F}$ for some $0 \leq \alpha \leq 1$.*

Proof. The $\widetilde{P}$ is bounded (by definition) and therefore a compact space. Let $P(\mathbf{A}, \mathbf{b})$ be the polyhedral representation of $\widetilde{P}$, which is the intersection of right- and left-closed half-spaces. Therefore, for $\forall \mathbf{x} \in \widetilde{P}$,

$$\alpha^* = \min_{0 \leq \alpha \leq 1} \{\alpha \mid \alpha \mathbf{x} \ \in \widetilde{P}\}$$

is defined uniquely. Let right-closed rows of the $P(\mathbf{A},\mathbf{b})$ (non-negative rows) be represented by $(\widehat{\mathbf{A}},\widehat{\mathbf{b}})$, then for any $\mathbf{x}\in\widetilde{P}$, α^* can be defined uniquely as following:

$$\alpha^* = max\left\{\frac{\widehat{\mathbf{b}}_i}{\widehat{\mathbf{A}}_{i,\cdot}\mathbf{x}}\right\}$$

where $\widehat{\mathbf{A}}_{i,\cdot}$ and $\widehat{\mathbf{b}}_i$ are the $i-$th row of the matrix $\widehat{\mathbf{A}}$ and vector $\widehat{\mathbf{b}}$ respectively.

This selection of α^* will guarantee that $(\alpha^*\mathbf{x})$ will satisfy at least one of the right-closed hyperplanes equation and also be in the feasible region defined by other right- and left-closed hyperplanes. Hence, for the aforementioned α^*, the $(\alpha^*\mathbf{x})$ satisfies all the constraints (rows) of $P(\mathbf{A},\mathbf{b})$, so $(\alpha^*\mathbf{x})\in P(\mathbf{A},\mathbf{b})=\widetilde{P}$, which suggests that there exist at least one right-closed facet of $\widetilde{P}$, $\mathcal{F}$, such that $\alpha^*\mathbf{x}\in\mathcal{F}$. □

Now we are ready to prove the following result.

Theorem 4.4.4. *A right-closed set $\mathcal{M}$ is convex if and only if $\forall\boldsymbol{x}\in\mathcal{F}_i$, where $\mathcal{F}_i$ is a member of set of right-closed facets defining the $\widetilde{P}$, $\lceil\boldsymbol{x}\rceil\in\mathcal{M}$.*

Proof. (Only if) Suppose $\exists\mathbf{x}\in\mathcal{F}_i$ $(\Rightarrow\mathbf{x}\in\widetilde{P})$ and $\lceil\mathbf{x}\rceil\notin\mathcal{M}$. There are two different scenarios:

1. either $\lceil\mathbf{x}\rceil$ is a member of $\widetilde{P}$ and as $\lceil\mathbf{x}\rceil\notin\mathcal{M}$, $\mathcal{M}$ is not convex, or

2. there is another minimal element of $\widetilde{P}$, $\tilde{\mathbf{m}}$, which is smaller than $\lceil\mathbf{x}\rceil$. $\tilde{\mathbf{m}}$ cannot belong to $\mathcal{M}$ either, because its membership to $\mathcal{M}$ implies $\lceil x\rceil\in\mathcal{M}$; Therefore $\mathcal{M}$ is not convex.

(If) Suppose $\mathcal{M}$ is not convex, then from the fact that $\min(Int(conv(\mathcal{M})))=\min(Int(\widetilde{P}))$, and theorem 4.2.7, $\exists\mathbf{x}\in\min(\widetilde{P})$ such that $\mathbf{x}\notin\mathcal{M}$. From theorem 4.4.3, $\exists\alpha^*$ such that $0\le\alpha^*\le1$, $\exists$ a facet $\mathcal{F}_i$ that is on a right-closed hyperplane of $\widetilde{P}$, such that $\alpha^*\mathbf{x}\in\mathcal{F}_i$. Since $\alpha^*\mathbf{x}\le\mathbf{x}$, $\mathbf{x}\notin\mathcal{M}$, and $\mathcal{M}$ is right-closed, it follows that $\alpha^*\mathbf{x}\notin\mathcal{M}$ as well; since $\lceil\alpha^*\mathbf{x}\rceil\le\alpha^*\mathbf{x}$, $\lceil\alpha^*\mathbf{x}\rceil\notin\mathcal{M}$ □

Essentially theorem 4.4.4 notes that

$$(\forall \text{ right-closed facets } \mathcal{F}_i \text{ of } \widetilde{P},\forall\mathbf{x}\in\mathcal{F}_i,\lceil\mathbf{x}\rceil\in\mathcal{M})\Leftrightarrow(\mathcal{M}\text{ is convex}). \tag{4.3}$$

This leads to a polynomial time algorithm for verifying the convexity of a right-closed set of integral vectors, which is presented in the next section.

4.5 Example of Section 4.3 Revisited

Earlier, we noted that each right-closed set $\mathcal{M}$ has a $\mathcal{V}$-polytope that is generated by members of $\min(\mathcal{M})$ and their elevated-counterparts. In the previous sections we introduced a $\widetilde{P}$-polytope that is essentially the smallest polytope that is defined by just right- and left-closed facet defining hyperplanes (i.e. every facet of $\widetilde{P}$ must be either be a right-closed facet or a left-closed facet) that contains the $\mathcal{V}$-polytope. This concept is reinforced by computing the $\widetilde{P}$-polytopes for the examples of section 4.3 this section.

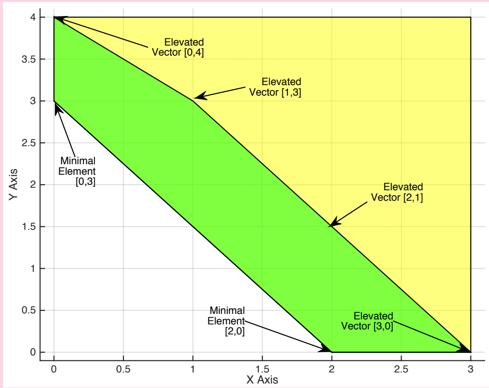
In figure 4.1 presented the $\mathcal{V}$-polytopes for two right-closed sets $\mathcal{M}_1$ and $\mathcal{M}_2$, where

$$\begin{aligned}\min(\mathcal{M}_1) &= \{(2\ 0)^T, (0\ 3)^T\}, \text{ and} \\ \min(\mathcal{M}_2) &= \{(0\ 4)^T, (1\ 3)^T, (2, 1)^T, (3\ 0)^T\}.\end{aligned}$$

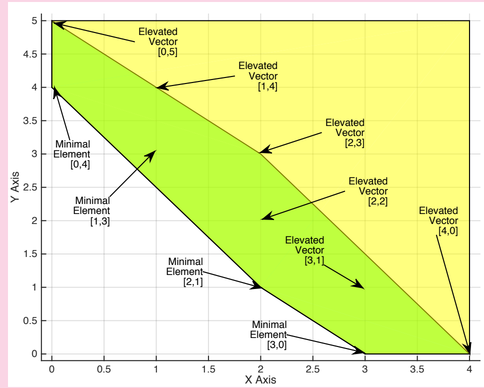
These polytopes along with their corresponding $\widetilde{P}$-polytopes are shown in figure 4.3(a) and 4.3(b), respectively. Each $\mathcal{V}$-polytope is shown in green, while its corresponding $\widetilde{P}$-polytope is shown in yellow in these figures. It is not hard to see that the yellow $\widetilde{P}$-polytope is the smallest polytope that is defined by just right- and left-closed facet defining hyperplanes that contains the green $\mathcal{V}$-polytope.

We considered a right-closed set $\mathcal{M}_3$ with minimal elements $A = (1\ 4\ 2)^T, B = (3\ 5\ 1)^T$ and $C = (5\ 1\ 3)^T$ in the discussion that accompanied figure 4.2. For this example, the $\mathcal{V}$-polytope is defined by the vertex set $\{(1\ 4\ 2)^T$, $(2\ 4\ 2)^T$, $(1\ 5\ 2)^T$, $(1\ 4\ 3)^T$, $(3\ 5\ 1)^T$, $(4\ 5\ 1)^T$, $(3\ 6\ 1)^T$, $(3\ 5\ 2)^T$, $(5\ 1\ 3)^T$, $(6\ 1\ 3)^T$, $(5\ 2\ 3)^T$, $(5\ 1\ 4)^T\}$. We had noted earlier that *Polymake* identifies ten facet defining hyperplanes in the polytope: $F1 : x + 3y + 5z \geq 23$, $F2 : -y - z \geq -7$, $F3 : x + 2z \geq 5$, $F4 : x \geq 1$, $F5 : z \geq 1$, $F6 : -x - 3y - 5z \geq -28$, $F7 : x + \frac{4}{3}y \geq \frac{19}{3}$, $F8 : y \geq 1$, $F9 : y + 2z \geq 7$, and $F10 : -x - y - z \geq -10$.

The polytope $\widetilde{P}$ for this example is identified by the right-closed facets of the polytope $\mathcal{V}$ (i.e.



(a) $conv(\mathcal{M}_1)$

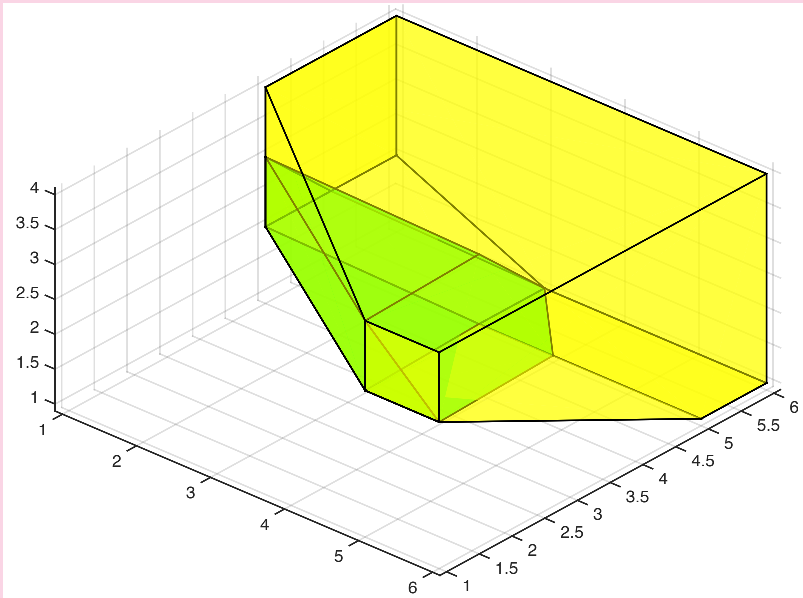


(b) $conv(\mathcal{M}_2)$

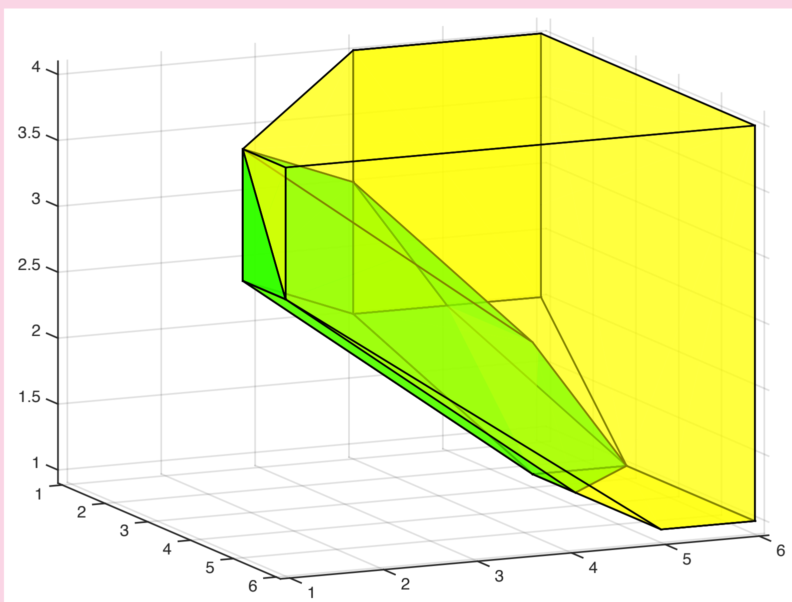
Figure 4.3: Illustration of the $\mathcal{V}$- and $\widetilde{P}$-polytopes for right-closed sets. The $\mathcal{V}$-polytope is shown in green, and the $\widetilde{P}$-polytope is shown in yellow. Figure (a) (resp. figure (b)) shows these polytopes for the right-closed set $\mathcal{M}_1$ (resp. $\mathcal{M}_2$) introduced in section 4.3. In each of these examples, we note: (1) there are facet defining hyperplaces in the green polytope that are neither right- nor left-closed, (2) each face defining hyperplane of the yellow polytope is either left- or a right-closed, and (3) the yellow polytope is the smallest polytope, described by right- and left-closed facet defining hyperplanes, that contains the green polytope.

$F1, F3, F4, F5, F7, F8$ and $F9$, in the list of facets defined above), together with three left-closed facets $G1 : x \leq 6, G2 : y \leq 6$ and $G3 : z \leq 4$. These left-closed facets are identified by the maximal-vector of the vertex set of $\mathcal{V}$, viz. $\max\{(1\ 4\ 2)^T$, $(2\ 4\ 2)^T$, $(1\ 5\ 2)^T$, $(1\ 4\ 3)^T$, $(3\ 5\ 1)^T$, $(4\ 5\ 1)^T$, $(3\ 6\ 1)^T$, $(3\ 5\ 2)^T$, $(5\ 1\ 3)^T$, $(6\ 1\ 3)^T$, $(5\ 2\ 3)^T$, $(5\ 1\ 4)^T\} = (6\ 6\ 4)^T$.

Figure 4.4 shows two different views of the polytopes $\mathcal{V}$ and $\widetilde{P}$ for this example. The polytope $\mathcal{V}$ is shown in green, while the polytope $\widetilde{P}$ is shown in yellow. It is not hard to see that the polytope $\widetilde{P}$ contains the polytope $\mathcal{V}$, and furthermore, every facet of $\widetilde{P}$ is either left- or right-closed facet. This observation plays a critical role in the convexity testing algorithms of the next chapter.



(a) Convex Hull - View 1



(b) Convex Hull - View 2

Figure 4.4: We revisit the example of figure 4.2, where we considered a right-closed set $\mathcal{M}_1$ with minimal elements $A = (1\ 4\ 2)^T, B = (3\ 5\ 1)^T$ and $C = (5\ 1\ 3)^T$. As noted earlier, the polytope $\mathcal{V}$ for this example is defined by the vertex set $\{(1\ 4\ 2)^T$, $(2\ 4\ 2)^T$, $(1\ 5\ 2)^T$, $(1\ 4\ 3)^T$, $(3\ 5\ 1)^T$, $(4\ 5\ 1)^T$, $(3\ 6\ 1)^T$, $(3\ 5\ 2)^T$, $(5\ 1\ 3)^T$, $(6\ 1\ 3)^T$, $(5\ 2\ 3)^T$, $(5\ 1\ 4)^T\}$. This polytope is shown in green in the above figure. The polytope $\widetilde{P}$, shown in yellow in the above figure, is the smallest polytope defined only by right- and left-closed facets (i.e. every facet of this polytope is either a left- or a right-closed facet), that contains the polytope $\mathcal{V}$.

Chapter 5

Convexity Testing Algorithms

In this chapter, we present two algorithms for testing the convexity of a right-closed integral set. The first algorithm is an *LP-Based* algorithm, which takes a polynomial time to test the convexity. The second one is based on the *grid-search* algorithm. Although not efficient as the first one, it still has a reasonable computational time and it can be easily implemented.

5.1 On verifying the condition of theorem 4.4.4

Verifying the condition mentioned in theorem 4.4.4 and equation 4.3 is not a feasible task. There are infinitely many points on each right-closed facet of the $\widetilde{P}$. To overcome this problem, we define a newer set over each right-closed facet of $\widetilde{P}$ to make the problem more tractable. Each right-closed facet $\mathcal{F}_i$ is essentially defined by a collection of vertices. Each facet-defining vertex is either some $\mathbf{m}_i \in \min(\mathcal{M})$; or some $\widetilde{\mathbf{m}}_i \geq \mathbf{m}_i$, for some $\mathbf{m}_i \in \min(\mathcal{M})$. Let

$$\Upsilon(\mathcal{F}_i) := \{\mathbf{m}_i \in \min(\mathcal{M}) \mid \text{Either } \mathbf{m}_i, \text{ or some } \widetilde{\mathbf{m}}_i \geq \mathbf{m}_i \text{ is a vertex that defines } \mathcal{F}_i\}. \tag{5.1}$$

The following result uses the set $\Upsilon(\mathcal{F}_i)$ in the test identified in theorem 4.4.4.

Theorem 5.1.1.

$$(\forall \textit{ right-closed facets } \widetilde{\mathcal{F}}_i \textit{ of } \widetilde{P}, \forall \widetilde{\mathbf{x}} \in conv(\Upsilon(\widetilde{\mathcal{F}}_i), \lceil \widetilde{\mathbf{x}} \rceil \in \mathcal{M}) \Leftrightarrow$$
$$(\forall \textit{ right-closed facets } \mathcal{F}_i \textit{ of } \widetilde{P}, \forall \mathbf{x} \in \mathcal{F}_i, \lceil \mathbf{x} \rceil \in \mathcal{M}).$$

Proof. ($\Rightarrow$ Part) Suppose $\exists$ a right-closed facet $\mathcal{F}_i$ of $\widetilde{P}$, and $\exists \mathbf{x} \in \mathcal{F}_i$, such that $\lceil \mathbf{x} \rceil \notin \mathcal{M}$. $\mathbf{x} \in \mathcal{F}_i$ can be written as the convex combination of the vertices for the given facet. By definition of the $\mathcal{F}_i$ in theorem 4.4.2, we can infer that a vertex of $\mathcal{F}_i$ is either a minimal elements of original right-closed set, or a vector $\mathbf{v}_i$ of the form: $\mathbf{v}_j = \mathbf{m}_j + \mathbf{y}_j$, for some $\mathbf{y}_j \in \mathcal{N}^n$ such that $\mathbf{y}_j \geq \mathbf{0}$. Hence,

$$\lceil \mathbf{x} \rceil = \left\lceil \sum_i \lambda_i \mathbf{m}_i + \sum_j \mu_j \mathbf{v}_j \right\rceil = \left\lceil \sum_i \lambda_i \mathbf{m}_i + \sum_j \mu_j (\mathbf{m}_j + \mathbf{y}_j) \right\rceil,$$

where $\sum_i \lambda_i + \sum_j \mu_j = 1$. If $\widetilde{\mathbf{x}} = \sum_i \lambda_i \mathbf{m}_i + \sum_j \mu_j \mathbf{m}_j$, then $\widetilde{\mathbf{x}} \in conv(\Upsilon(\mathcal{F}_i))$, $\widetilde{\mathbf{x}} \leq \mathbf{x} (\Rightarrow \lceil \widetilde{\mathbf{x}} \rceil \leq \lceil \mathbf{x} \rceil)$, and since $\mathcal{M}$ is right-closed, $\lceil \widetilde{\mathbf{x}} \rceil \notin \mathcal{M}$.

($\Leftarrow$ Part) Suppose $\exists$ a right-closed facet $\widetilde{\mathcal{F}}_i$ of $\widetilde{P}$, and $\exists \widetilde{\mathbf{x}} \in \widetilde{\mathcal{F}}_i$, such that $\lceil \widetilde{\mathbf{x}} \rceil \notin \mathcal{M}$. Since $\mathcal{M}$ is right-closed, the set of integral vectors in $conv(\mathcal{M})$ is also right-closed (cf. Lemma III.5, [25]). Additionally, $\widetilde{\mathbf{x}} \in conv(\mathcal{M})$, $\widetilde{\mathbf{x}} \leq \lceil \widetilde{\mathbf{x}} \rceil (\Rightarrow \lceil \widetilde{\mathbf{x}} \rceil \in conv(\mathcal{M}))$, and since $\lceil \widetilde{\mathbf{x}} \rceil \notin \mathcal{M}$, it follows that $\mathcal{M}$ is not convex. From theorem 4.4.4, $\exists$ a right-closed facet $\mathcal{F}_i$ of $\widetilde{P}$, and $\exists \mathbf{x} \in \mathcal{F}_i$, such that $\lceil \mathbf{x} \rceil \notin \mathcal{M}$. $\square$

As a consequence of theorem 5.1.1, equation 4.3 can be written equivalently as

$$(\forall \text{ right-closed facets } \widetilde{\mathcal{F}}_i \text{ of } \widetilde{P}, \forall \widetilde{\mathbf{x}} \in conv(\Upsilon(\widetilde{\mathcal{F}}_i)), \lceil \widetilde{\mathbf{x}} \rceil \in \mathcal{M}) \Leftrightarrow (\mathcal{M} \text{ is convex}). \tag{5.2}$$

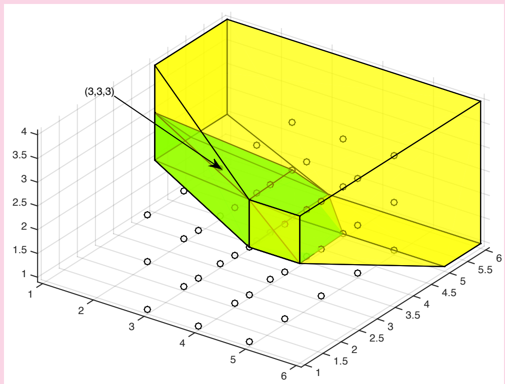
This observation leads to the following corollary.

Corollary 1. *If $\mathcal{M} \subseteq \mathcal{N}^n$ is a right-closed set such that $\min(\mathcal{M}) \subseteq \{0,1\}^n$, then $\mathcal{M}$ is convex.*

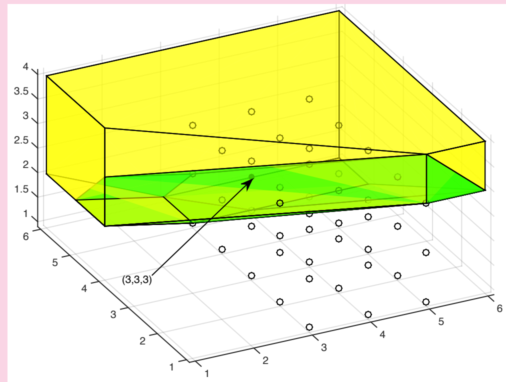
We present two efficient procedures that checks the condition $(\forall \widetilde{\mathbf{x}} \in conv(\Upsilon(\mathcal{F}_i)), \lceil \widetilde{\mathbf{x}} \rceil \in \mathcal{M})$ in the next two subsections. The first method, yields a polynomial time algorithm for verifying convexity of a right-closed set.

5.1.1 LP- based Algorithm

The presented test condition requires us to find the ceiling function for the convex combination of the members of $\Upsilon(\mathcal{F}_i)$. Instead of going through all the points inside the convex hull of $\Upsilon(\mathcal{F}_i)$, one approach can be creating a "box" over each right-closed facet; Basically, we can create a set of integer vectors "proximal" to the right-closed facet, and then verify their representation as a ceiling of convex combination of minimal elements, by a feasibility LP. Figure 5.1 clarifies this approach.



(a) Creating the "box" (View 1)



(b) Creating the "box" (View 2)

Figure 5.1: Illustration of LP-based algorithm - check if the integral vectors in the "box" are proximal to the right-closed facet indicated above. Vector $(3\ 3\ 3)^T$, shown above, is the only integral vector that is proximal to the right-closed facet identified above.

We first present the feasibility LP and then describe the procedure to create the mentioned box, called $\mathcal{L}(\mathcal{F}_i)$. Note that the following theorem establishes the polynomial time solvability of a key membership question that is used to test the condition of theorem 5.1.1.

Theorem 5.1.2. *Let* $\Upsilon(\mathcal{F}_i) = \{\boldsymbol{m}_1.\boldsymbol{m}_2, \ldots, \boldsymbol{m}_k\}$ *for a right-closed facet* $\mathcal{F}_i$ *of* $\widetilde{P}$*, and* $\widetilde{\mathbf{m}} \in \mathcal{N}^n$ *be an integral vector. Then,* $\widetilde{\mathbf{m}} = \lceil \sum_i^k \lambda_i \boldsymbol{m}_i \rceil$ *for a set* $\{\lambda_i\}_{i=1}^k$ *such that* $\forall i \in \{1, \ldots, k\}, 0 \leq \lambda_i \leq 1$ *and*

$\sum_{i=1}^{k}\lambda_i = 1$, *if and only if the Linear Program (LP) returns an optimal value* $\alpha^* \neq 0$.

$$LP(\widetilde{\boldsymbol{m}}) : \max(\alpha)$$

$$\textit{subject to}$$

$$\begin{pmatrix} \boldsymbol{m}_1 & \boldsymbol{m}_2 & \ldots & \boldsymbol{m}_k \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \end{pmatrix} \geq (\widetilde{\boldsymbol{m}} - \boldsymbol{1}) + (\alpha \times \boldsymbol{1})$$

$$\begin{pmatrix} \boldsymbol{m}_1 & \boldsymbol{m}_2 & \ldots & \boldsymbol{m}_k \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \end{pmatrix} \leq \widetilde{\boldsymbol{m}}$$

$$\sum_{i=1}^{k}\lambda_i = 1, \forall i \in \{1, \ldots, k\}, 0 \leq \lambda_i \leq 1, \ \textit{and}\ 0 \leq \alpha,$$

where $\mathbf{1} \in \mathcal{N}^n$ *is the vector of all ones.*

Proof. (*Only If*) Suppose $\widetilde{\mathbf{m}} = \lceil \sum_{i}^{k} \lambda_i \mathbf{m}_i \rceil$ for a set $\{\lambda_i\}_{i=1}^{k}$ such that $\forall i \in \{1, \ldots, k\}, 0 \leq \lambda_i \leq 1$ and $\sum_{i=1}^{k}\lambda_i = 1$. Then it follows that $\widetilde{\mathbf{m}} - \mathbf{1} < \sum_{i}^{k} \lambda_i \mathbf{m}_i \leq \widetilde{\mathbf{m}}$. It follows that

$$\alpha^* \geq \min_j \left\{ \left(\sum_{i=1}^{k} \lambda_i \mathbf{m}_i \right)_j - (\widetilde{\mathbf{m}}_j - 1) > 0, \right\}$$

where $(\bullet)_j$ denotes the j-th component of a vector argument.

(*If*) From the constraints of LP, it follows that $\alpha^* \leq 1$. If $\alpha^* > 0$ with a corresponding set – $\{\lambda_i\}_{i=1}^{k}$ – that is feasible for LP, we can infer that

$$\widetilde{\mathbf{m}} - \mathbf{1} < \sum_{i=1}^{k} \lambda_i \mathbf{m}_i \leq \widetilde{\mathbf{m}} \Rightarrow \left\lceil \sum_{i=1}^{k} \lambda_i \mathbf{m}_i \right\rceil = \widetilde{\mathbf{m}}.$$

$\square$

For each facet $\mathcal{F}_i$ of $\widetilde{P}$, let us define

$$\mathcal{L}(\mathcal{F}_i) := \{\mathbf{x} \in \mathcal{N}^n \mid \exists \mathbf{y} \in \mathcal{F}_i \text{ where } \mathbf{x} = \lceil \mathbf{y} \rceil\}.$$

From theorem 5.1.1 the right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$ is convex if and only if $\mathcal{L}(\mathcal{F}_i) - \mathcal{M} = \emptyset$. For each member $\widetilde{\mathbf{m}}$ of $\mathcal{L}(\mathcal{F}_i)$, theorem 5.1.2 yields a polynomial time algorithm that decides if $\widetilde{\mathbf{m}} \in \mathcal{M}$. Now, we address the issue of estimating an upper bound on the size of $\mathcal{L}(\mathcal{F}_i)$.

Let:

$$\epsilon_i := max(\mathbf{m}_{1_i}, \mathbf{m}_{2_i}, \ldots, \mathbf{m}_{k_i}) - min(\mathbf{m}_{1_i}, \mathbf{m}_{2_i}, \ldots, \mathbf{m}_{k_i}),$$

and $c = \max_i(\epsilon_i)$, then it follows that $card(\mathcal{L}(\mathcal{F}_i)) \leq c^n$. The quantity c can be interpreted as a measure-of-variation among the individual components of the members of $\Upsilon(\mathcal{F}_i)$. For a fixed dimension (i.e. n is fixed), it follows that the size of $\mathcal{L}(\mathcal{F}_i)$ is polynomial in the measure-of-variation c. We argue that the set $\mathcal{L}(\mathcal{F}_i)$ can be constructed in polynomial time.

Let $\beta_i = \min(\mathbf{m}_{1_i}, \ldots, \mathbf{m}_{k_i})$, and

$$\mathcal{S}(\mathcal{F}_i) := \{\mathbf{x} \in \mathcal{N}^n \mid \forall i \in \{1, \ldots, n\}, \mathbf{x}_i \in [\beta_i, \beta_i + c]\} \tag{5.3}$$

$\mathcal{S}(\mathcal{F}_i)$ is the superset of $\mathcal{L}(\mathcal{F}_i)$, where $card(\mathcal{S}(\mathcal{F}_i)) = c^n$. Hence,

$$\forall \mathbf{x} \in \mathcal{S}(\mathcal{F}_i) \text{ if } \text{LP}(\mathbf{x}) > 0 \Rightarrow \mathbf{x} \in \mathcal{L}(\mathcal{F}_i)$$

As a consequence of the polynomial time solvability of LPs, this condition can be checked in polynomial time.

5.1.2 Convexity Testing Algorithm

Algorithm 4 tests the convexity of a right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$. Its correctness follows directly from the results in this thesis.

Algorithm 4 Testing Convexity Algorithm

Compute $\widetilde{P}$ (cf. equation 4.2 and theorem 4.4.2).
for Each right-closed facet $\mathcal{F}_i$ of $\widetilde{P}$ **do**
    Compute $\Upsilon(\mathcal{F}_i) = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_k\}$ (cf. equation 5.1)
    for Each $i \in \{1, 2, \ldots, n\}$ **do**
        Compute $\epsilon_i =: max(\mathbf{m}_{1_i}, \mathbf{m}_{2_i}, \ldots, \mathbf{m}_{k_i}) - min(\mathbf{m}_{1_i}, \mathbf{m}_{2_i}, \ldots, \mathbf{m}_{k_i})$.
        $c = \max_i(\epsilon_i)$.
        Compute $\mathcal{S}(\mathcal{F}_i)$ (cf. equation 5.3)
        $\mathcal{L}(\mathcal{F}_i) = \varnothing$
        for Every element $\mathbf{x} \in \mathcal{S}(\mathcal{F}_i)$ **do**
            if $(\mathrm{LP}(\mathbf{x}) > 0) \wedge (\mathbf{x} \ngeq \widetilde{\mathbf{m}}), \forall\, \widetilde{\mathbf{m}} \in \mathcal{L}(\mathcal{F}_i)$ **then**
                $\mathcal{L}(\mathcal{F}_i) = \mathcal{L}(\mathcal{F}_i) \cup \{\mathbf{x}\}$
        for Every $\widetilde{\mathbf{m}} \in \mathcal{L}(\mathcal{F}_i)$ **do**
            if $\widetilde{m} \notin \mathcal{M}$ **then**
                Not convex; Exit.
Convex; Exit.

The first step in the algorithm 4 is computing $\widetilde{P}$. From the discussion following theorem 4.4.2, this can be accomplished in polynomial time. By theorem 4.2.4, such a procedure for m points in n-dimension, can be done deterministically in $\mathcal{O}(mlog(m) + m^{\lfloor n/2 \rfloor})$; when n is fixed, this is a polynomial time operation. It is worth mentioning that we assume $card(\min(\mathcal{M})) = m$. Next step calculates the superset $\mathcal{S}(\mathcal{F}_i)$, which can be done in $\mathcal{O}(m)$ time. For each $\mathbf{x} \in \mathcal{S}(\mathcal{F}_i)$, we need to compute $LP(\mathbf{x})$, which can be accomplished in polynomial time as well. This procedure has to be done $card(\mathcal{S}(\mathcal{F}_i)) = c^n$ times, where $c = \max_i(\epsilon_i)$ is a fixed number. Therefore, calculating $\mathcal{L}(\mathcal{F}_i)$, also can be done in polynomial time. Note that $\mathcal{L}(\mathcal{F}_i)$, contains only the minimal elements of $\mathcal{L}(\mathcal{F}_i)$, which can reduce the computational time further. The last step, verifying the membership of elements in $\mathcal{L}(\mathcal{F}_i)$ to $\mathcal{M}$ will take $\mathcal{O}(m)$ time. Since the number of facets for $\widetilde{P}$ is bounded above by $\mathcal{O}(m^{\lfloor n/2 \rfloor})$, algorithm 4 will only execute polynomial number of operations, giving us a polynomial time algorithm in m and c for testing the convexity of a right-closed set, when n is fixed.

5.1.3 Algorithm 4 and the Examples of Section 4.3

Figures 4.3(a) and 4.3(b) presented the $\widetilde{P}$-polytope for the right-closed sets $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. There is only one right-closed facet in the polytope(s) for $\mathcal{M}_1$, and this facet contains the minimal elements $(0\ 3)^T$ and $(2\ 0)^T$. There is a lattice point $(1\ 2)^T$ that is "proximal" to this facet (i.e. there is no other lattice point in the polytope that is term-wise smaller and closer to the facet; cf. figure 5.2(a)). The linear programming formulation in theorem 5.1.2 tests if a vector is proximal to a right-closed facet $\mathcal{F}_i$, when this formulation is used in the present example, we note

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} = \left\lceil 0.6 \times \begin{pmatrix} 0 \\ 3 \end{pmatrix} + 0.4 \times \begin{pmatrix} 2 \\ 0 \end{pmatrix} \right\rceil = \left\lceil \begin{pmatrix} 0.8 \\ 1.8 \end{pmatrix} \right\rceil,$$

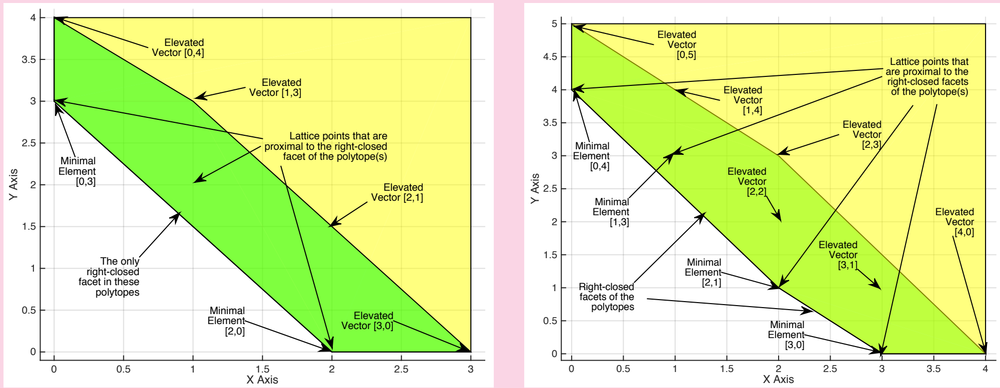
and $LP((1\ 2)^T) = 0.8 > 0$, as expected. Since $(1\ 2)^T \notin \mathcal{M}_1$, algorithm 4 will conclude with the pronouncement that $\mathcal{M}_1$ is not convex.

In contrast consider the $\widetilde{P}$-polytope for the right-closed set $\mathcal{M}_2$ of section 4.3, which is shown in figure 5.2(b). There are four lattice points, $\{(0\ 4)^T, (1\ 3)^T, (2\ 1)^T, (3\ 0)^T\}$, that are proximal to the two right-closed facets of this polytope (cf. figure 5.2(b)). Since each of these vectors are in $\mathcal{M}_2$, algorithm 4 will declare that $\mathcal{M}_2$ is convex.

In the example of figure 4.4, where we considered a right-closed set set $\mathcal{M}_3$ with minimal elements $A = (1\ 4\ 2)^T, B = (3\ 5\ 1)^T$ and $C = (5\ 1\ 3)^T$. The polytope $\widetilde{P}$, shown in yellow, contains the right-closed facet $F1 : x + 3y + 5z = 23$, and $\Upsilon(F1) = \{A, B, C\}$. If $\widetilde{\mathbf{m}} = (3\ 3\ 3)^T$, we get $LP(\widetilde{\mathbf{m}}) = 0.5$. Theorem 5.1.2 is verified by noting that

$$\begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix} = \left\lceil \frac{1}{2} \times \begin{pmatrix} 5 \\ 1 \\ 3 \end{pmatrix} + \frac{1}{2} \times \begin{pmatrix} 1 \\ 4 \\ 2 \end{pmatrix} \right\rceil = \left\lceil \begin{pmatrix} 3 \\ 2.5 \\ 2.5 \end{pmatrix} \right\rceil.$$

Since $(3\ 3\ 3)^T \notin \mathcal{M}_1$, we conclude that $\mathcal{M}_1$ is not convex, which is confirms earlier findings, as well. Figure 5.3 shows the location of the right-closed facet $F1$ in the $\mathcal{V}$- and $\widetilde{P}$-polytopes for this example



(a) $conv(\mathcal{M}_1)$

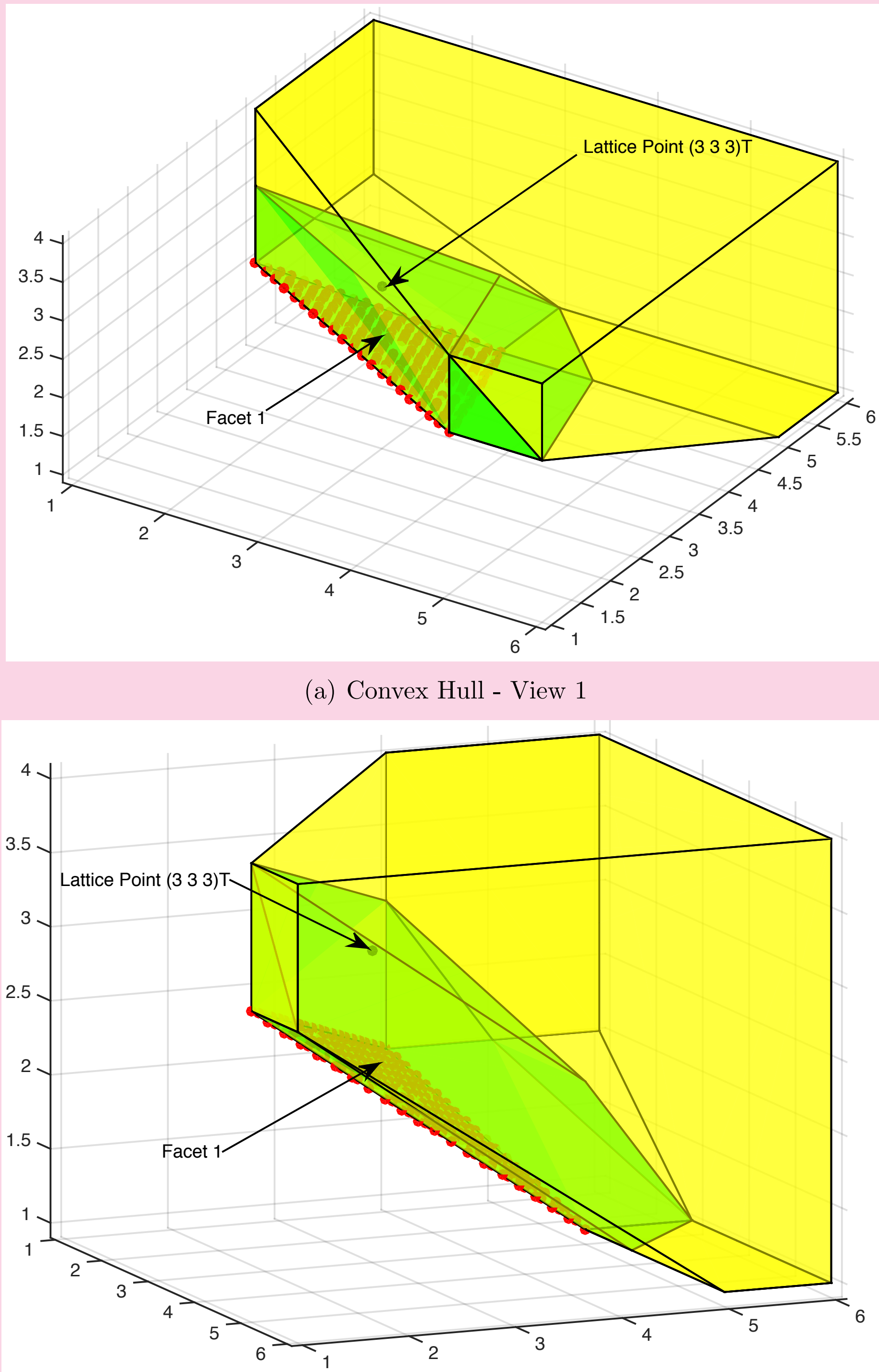
(b) $conv(\mathcal{M}_2)$

Figure 5.2: This figure illustrates the fact that the LP-formulation of theorem 5.1.2 is a test if a vector is *proximal* to a right-closed facet $\mathcal{F}_i$. A proximal lattice point in a polytope is essentially an integral vector that is closest to a facet of the polytope. Figure (a) shows that there is a lattice point, $(1\ 2)^T$, that is proximal to a right-closed facet, that is not in $\mathcal{M}_1$. Figure (b) shows that all proximal lattice points to right-closed facets are in $\mathcal{M}_2$.

along with the location of the $(3\ 3\ 3)^T$ vector that is proximal to this facet. Algorithm 4 would consequently declare $\mathcal{M}_3$ be non-convex as expected.

5.2 Grid Search Algorithm

One other approach for verifying the condition mentioned in theorem 4.4.4 is to select the values of λ which makes the change in ceiling function. We are looking at the ceiling function of convex combination of the points inside the $\Upsilon(\mathcal{F}_i)$. The ceiling function has a step-like property and changes only for countably many values of λ's. Therefore, there is no need to search for all values between 0 and 1 on a real interval; Selecting a proper step size for λ's on this interval, can guarantee that all the possible points will be explored. Second proposed algorithm is a grid-search algorithm with a proper step size for the convex combination multiplier.



(a) Convex Hull - View 1

(b) Convex Hull - View 2

Figure 5.3: We revisit the example of figure 4.4, where we considered a right-closed set $\mathcal{M}_1$ with minimal elements $A = (1\ 4\ 2)^T, B = (3\ 5\ 1)^T$ and $C = (5\ 1\ 3)^T$. As noted earlier, the polytope $\mathcal{V}$ (resp. $\widetilde{P}$) is shown in green (resp. yellow) in the above figure. Facet $F1 : x + 3y + 5z = 23$ contains the vertices A, B and C. The grid-search procedure on facet $F1$, identified by the red-points in the above views, finds the integral vector $(3\ 3\ 3)^T$ which is proximal to $F1$ that is not in $\mathcal{M}_1$.

5.2.1 Step Size for λ

Theorem 5.1.1 requires us to check the membership of ceiling function of all possible convex combination of $\Upsilon(\mathcal{F}_i)$, for every right-closed facet, $\mathcal{F}_i$ in set $\mathcal{M}$. Implementing such a test condition is not possible. Thus, we need to modify our search space to a finite set. The best candidate to such a modification is the convex combination coefficients. The ceiling function is a step-like function which its value changes only at some *finite* points. Therefore, instead of evaluating the λ_i's over a continuous interval of $[0, 1]$, we can perform a grid-search with step size of ϵ^*. We can define $\epsilon_i =: (\max(\mathbf{m}_{1_i}, \mathbf{m}_{2_i}, \ldots, \mathbf{m}_{k_i}) - \min(\mathbf{m}_{1_i}, \mathbf{m}_{2_i}, \ldots, \mathbf{m}_{k_i}))$, for each $i \in \{1, 2, \ldots, n\}$, and then assign each $\lambda_j, j \in \{1, 2, \ldots, k\}$, on of the $\epsilon^* = (\epsilon_1\epsilon_2 \ldots \epsilon_n)$-many values of $\frac{l}{\epsilon^*}$ with regard to constraints of $\sum_j \lambda_j = 1$ and $l \in \{0, 1, \ldots, \epsilon^*\}$. Although feasible to perform, still the number of operations to be done in this case is very large. By slight modification in the ϵ_i's definition we can greatly reduce the search space. By re-arranging the test condition for $\Upsilon(\mathcal{F}_i) = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_k\}$, we can write:

$$\lceil \mathbf{x} \rceil = \lceil \lambda_1\mathbf{m}_1 + \lambda_2\mathbf{m}_2 + \ldots + \lambda_{k-1}\mathbf{m}_{k-1} + (1 - \lambda_1 - \lambda_2 - \cdots - \lambda_{k-1})\mathbf{m}_k \rceil$$

$$= \left\lceil \sum_{i=1}^{k-1} \lambda_i(\mathbf{m}_i - \mathbf{m}_k) \right\rceil + \mathbf{m}_k$$

where $0 \leq \lambda_i \leq 1$, and $\sum_{i=1}^{k-1} \lambda_i \leq 1$. Now, by slight modification we define $\epsilon_i =: (max((\mathbf{m}_{1_i} - \mathbf{m}_{k_i}), (\mathbf{m}_{2_i} - \mathbf{m}_{k_i}), \ldots, (\mathbf{m}_{k-1_i} - \mathbf{m}_{k_i})) - min((\mathbf{m}_{1_i} - \mathbf{m}_{k_i}), (\mathbf{m}_{2_i} - \mathbf{m}_{k_i}), \ldots, (\mathbf{m}_{k-1_i} - \mathbf{m}_{k_i})))$. Note that each of the above elements are considered to be the absolute value of the difference. Then, we define $\epsilon^* = \prod_{i=1}^{n} \epsilon_i$. Hence, $\lambda_j = \frac{n_j}{\epsilon^*}$ for $n_j = \{0, 1, 2, \ldots, \epsilon^*\}$, such that: $\sum_{j=1}^{k-1} \lambda_j \leq 1$

For computing the step size for λ_j's over each right-closed facet, the first step will be the finding of $\mathbf{m}_k$ as the base point. To reduce the search space even more, one can easily find a $\mathbf{m}_k$ by brute-force, which produces the smallest ϵ^*. This point should be a point such that it's distance to all point is equal. We propose following optimization model, for finiding such a point. Note that the solution to the optimization problem is not necessarily the desired point but it can give a good estimate of such a point.

$$Min \sum_{i}^{n} \parallel \mathbf{m}_k - \mathbf{m}_i \parallel_2, \forall m_k \in \Upsilon(\mathcal{F}_i)$$

We will show that even choosing an arbitrary $\mathbf{m}_k$ will not result in an exponential number of choices. If $\lambda_i = \frac{n_i}{\epsilon^*}$ where $n_i \leq \epsilon^*$, then

$$\sum_{i=1}^{k-1} \lambda_i \leq 1 \Rightarrow \sum_{i=1}^{k-1} n_i \leq \epsilon^*$$

Then the search space for all possible values for $(n_1, n_2, \ldots, n_{k-1})$ is significantly smaller than $(\epsilon^*)^{k-1}$. This is very loose upper bound and in the following discussion we will tighten this bound even further for the number of possible values for such a selection.

In general, we are looking to number of possible choice for $\sum_{i=1}^{n} \mathbf{x}_i \leq c$ where $\forall i \in \{1, 2, \ldots, n\}, \mathbf{x}_i \geq 0$. Basically we are looking at number of integer points in the bounded polyhedron defined by $\sum_{i=1}^{n} \mathbf{x}_i \leq c$ with respect to the mentioned constraints. The next theorem deal with computing number of possible selection.

Theorem 5.2.1. *The number of lattice point inside the bounded polyhedron defined by* $\sum_{i=1}^{n} \boldsymbol{x}_i \leq c$ *where* $\forall i \in \{1, 2, \ldots, n\}, \boldsymbol{x}_i \geq 0$ *is defined by function* $f(c)$ *which is computed as:*

$$f(c) = \sum_{j=0}^{c} \binom{n+j-1}{j}$$

Proof. let $\Phi_{\mathbf{A}}(\mathbf{b}) = card(\{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0, \mathbf{x} \text{ is integral }\})$. The generating function for $\Phi_{\mathbf{A}}(\mathbf{b})$ when $\mathbf{A} = (1, 1, \ldots, 1)$, can be computed as following [42]:

$$\sum_{0}^{\infty} \Phi_{\mathbf{A}}(\mathbf{b}) t^b = \frac{1}{(1-t)^n}$$

Using the following *binomial series* identity (section 2.3 in [36]):

$$\frac{1}{(1-t)^{d+1}} = \sum_{k=0}^{\infty} \binom{d+k}{d} t^k$$

By replacing $\frac{1}{(1-t)^n}$ in former expression with the mentioned identity, we infer $\Phi_{\mathbf{A}}(\mathbf{b}) = \binom{n+b-1}{b}$. As we are dealing with the polyhedron $\sum_{i=1}^{n} \mathbf{x}_i \leq c$, with inequality, hence:

$$f(c) = \sum_{j=0}^{c} \binom{n+j-1}{j}$$

□

We can show that as n grows, the value of $f(c)$ is dominated by c^n completely.

Corollary 2. *For any $c > 1$,*

$$\lim_{n\to\infty} \frac{f(c)}{c^n} = 0$$

Proof. The term $f(c) = \sum_{j=0}^{c} \binom{n+j-1}{j}$ which is bounded above by $(c+1)\binom{(n+c-1)}{c}$. Therefore:

$$\lim_{n\to\infty} \frac{f(c)}{c^n} \leq \lim_{n\to\infty} \frac{(c+1)\binom{(n+c-1)}{c}}{c^n} \leq \lim_{n\to\infty} \frac{(c+1)\ n^c}{c^n\ c!}$$

c^n and c ! will grow much faster than n^c and $(c+1)$ receptively, as $n \to \infty$. Therefore:

$$\lim_{n\to\infty} \frac{f(c)}{c^n} = 0$$

□

The corollary shows that even choosing an arbitrary $\mathbf{m}_k$ for computing the step-size for λ_i, will not yield in an exponential number of selection. The figure 5.4 shows this comparison. As n grows, the $f(c)$ is dominated by c^n. This is true, even for small n. This means that the grid search, although not polynomial, but still is dominated by a exponential function in c, which in practice, does not seem to be an inefficient algorithm

Among all the possible choices of λi's, there are some values for which they will generate a similar value for the ceiling function mentioned in the test. By eliminating these values, we can reduce the search space even more. Each time, we evaluate $\lambda_i = \frac{n_i}{c}$ which essentially translates to searching over a grid of the type: $(n_1, n_2, \ldots, n_{k-1}) \in \mathcal{N}^n$, $\forall i$, $0 \le n_i \le c$ and $\sum_{i=1}^{k-1} \le c$. Each points in such a grid, effectively identifies an integral vector according to:

$$\mathbf{x} = \left\lceil \sum_{i=1}^{k-1} \frac{n_i}{c}(\mathbf{m}_i - \mathbf{m}_k) \right\rceil + \mathbf{m}_k$$

With a slight abuse of notation, we say $(n_1, n_2, \ldots, n_{k-1}) \in \mathcal{M}$ if and only if $\mathbf{x} \in \mathcal{M}$. The following corollary discusses cases in which we can eliminate the redundant grid points.

Corollary 3. *Suppose* $(\hat{n_1}, \hat{n_2}, \ldots, \hat{n_{k-1}}) \in \mathcal{N}^n$ *is a lattice point such that for each* $i \in \{1, 2, \ldots, n\}$, (1) $\hat{n_i} \ge n_i$ *and* $(\boldsymbol{x} - \boldsymbol{m}_i)_i \ge 0$ *or* (2) $\hat{n_i} \le n_i$ *and* $(\boldsymbol{x} - \boldsymbol{m}_i)_i \le 0$*; If* $(n_1, n_2, \ldots, n_{k-1}) \in \mathcal{M}$ *then* $(\hat{n_1}, \hat{n_2}, \ldots, \hat{n_{k-1}}) \in \mathcal{M}$ *as well.*

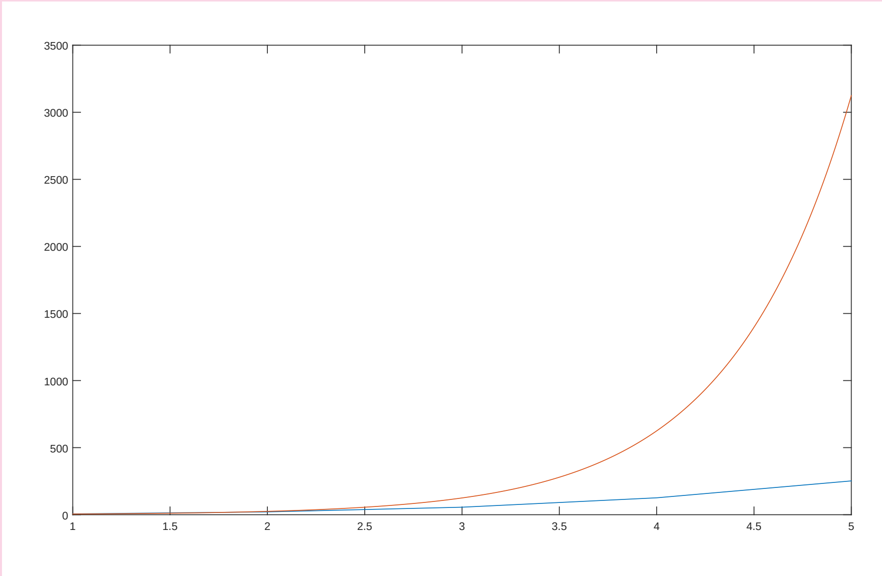
Proof. The proof is straight forward. If $(n_1, n_2, \ldots, n_{k-1}) \in \mathcal{M}$ and the first condition is true, then:

$$\hat{\mathbf{x}} = \left\lceil \sum_{i=1}^{k-1} \frac{\hat{n_i}}{c}(\mathbf{m}_i - \mathbf{m}_k) \right\rceil + \mathbf{m}_k \ge \mathbf{x}$$

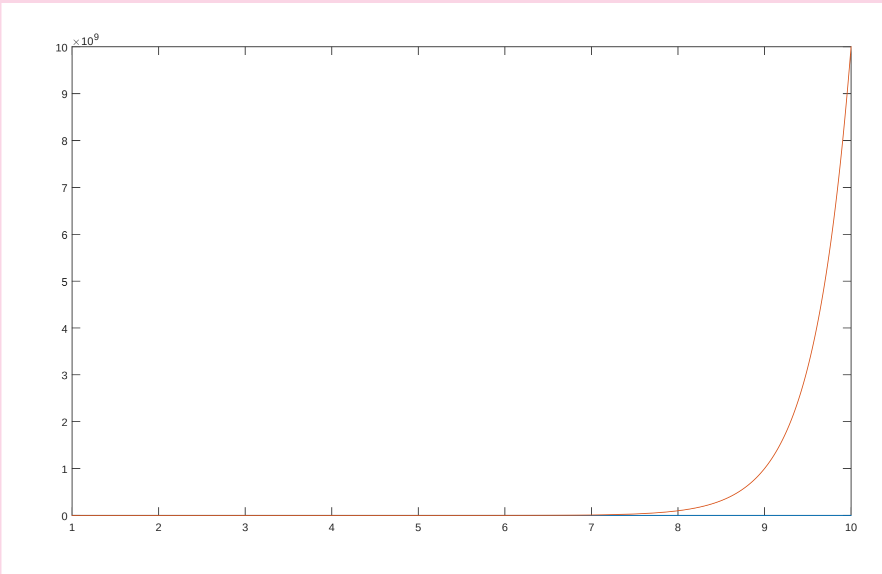
As the set $\mathbf{x} \in \mathcal{M}$ and $\mathcal{M}$ is right-closed, therefore $\hat{\mathbf{x}} \in \mathcal{M}$. By the same argument, if condition (2) holds, $\hat{\mathbf{x}} \ge \mathbf{x}$ and $\hat{\mathbf{x}} \in \mathcal{M}$ □

On the other hand, slight modification of conditions in the previous corollary, can also eliminated more unnecessary grid points in the search space. The proof will be similar as the previous one.

Corollary 4. *Suppose* $(\hat{n_1}, \hat{n_2}, \ldots, \hat{n_{k-1}}) \in \mathcal{N}^n$ *is a lattice point such that for each* $i \in \{1, 2, \ldots, n\}$, (1) $\hat{n_i} \ge n_i$ *and* $(\boldsymbol{x} - \boldsymbol{m}_i)_i \le 0$ *or* (2) $\hat{n_i} \le n_i$ *and* $(\boldsymbol{x} - \boldsymbol{m}_i)_i \ge 0$*; If* $(n_1, n_2, \ldots, n_{k-1}) \notin \mathcal{M}$ *then* $(\hat{n_1}, \hat{n_2}, \ldots, \hat{n_{k-1}}) \notin \mathcal{M}$ *as well.*



(a) $f(c)$ and c^n comparison- small n



(b) $f(c)$ and c^n comparison- large n

Figure 5.4: The c^n is dominant to $f(c)$ function. In figure (a), although the n is so small, the c^n is very dominant. As the n grows in figure (b), the result of corollary 2 is more clear.

Chapter 6

Discussion

As we proposed in the previous chapter, the *LP-based* algorithm can verify the convexity of a right-closed set in a polynomial time. Although efficient, the performance time can still be shortened. In this chapter, we are aiming for a heuristic-type algorithm which can test the convexity of right-closed set in much shorter time. This heuristic ables us to at least rule out the convexity for many instances. This result is followed by an illustrative example which shows how this heuristic method can help us verifying the convexity in a much shorter time. Lastly, we discuss an example on how to implement the monitor placement paradigm on PNs.

6.1 Heuristic Methods

The vector $\mathbf{x} \in \mathcal{N}^n$ is *lexicographically smaller* than $\mathbf{y} \in \mathcal{N}^n$, if $\exists k > 0$, such that $\forall i < k, \mathbf{x}_i = \mathbf{y}_i$ and $\mathbf{x}_k < \mathbf{y}_k$. We use $\mathbf{x} <^d \mathbf{y}$ to denote the fact that $\mathbf{x}$ is lexicographically smaller than $\mathbf{y}$. We say $\mathbf{x} <^d \mathbf{y}$ are *consecutive elements* of $\mathcal{M}$ if $\nexists\, \mathbf{z} \in \mathcal{M}$ such that $\mathbf{x} <^d \mathbf{z} <^d \mathbf{y}$.

Given two polytopes P_1 and P_2, Bemporad et al [43] show that $P_1 \cup P_2$ is convex if and only if for each vertex v_1 of P_1, and each vertex v_2 of P_2, the line segment $[v_1, v_2]$ is contained in $P_1 \cup P_2$. However, this result does not yield a necessary and sufficient condition for the union of two convex right-closed sets of integral vectors to be convex. A right-closed set with just one minimal element is convex, and the following lemma identifies a sufficient condition when the union of two such right-closed sets is convex.

Lemma 6.1.1. *Let $\mathbf{m}_1 \in \mathcal{M}$ and $\mathbf{m}_2 \in \mathcal{M}$ be two consecutive minimal elements of a right-closed set $\mathcal{M} \subseteq \mathcal{N}^n$. If the two sub-vectors obtained by removing/deleting each component in the set $\{i \in \{1, 2, \ldots, n\} \mid \; \mid \mathbf{m}_{1_i} - \mathbf{m}_{2_i} \mid = 1\}$ from $\mathbf{m}_1$ and $\mathbf{m}_2$, can be compared completely under the regular* *ordering, then the set* $\widehat{\mathcal{M}} = \{\mathbf{m} \in \mathcal{N}^n \mid (\mathbf{m} \geq \mathbf{m}_1) \vee (\mathbf{m} \geq \mathbf{m}_2)\}$ *is convex.*

Proof. By removing/deleting each j-th component, where $j \in \{i \in \{1,2,\ldots,n\} \mid \mid \mathbf{m}_{1_i} - \mathbf{m}_{2_i} \mid = 1\}$ of $\mathbf{m}_1$ and $\mathbf{m}_2$, we create sub-vectors $\widehat{\mathbf{m}}_1$ and $\widehat{\mathbf{m}}_2$. By assumption $\widehat{\mathbf{m}}_1$ and $\widehat{\mathbf{m}}_2$ can be compared completely under the regular ordering. Without loss of generality, let us suppose $\widehat{\mathbf{m}}_1 \leq \widehat{\mathbf{m}}_2$. Let us suppose we have a collection of vectors $\{\widetilde{\mathbf{m}}_i\}_{i=1}^p \subseteq \widehat{\mathcal{M}}$ such that $\mathbf{m}_3 = \sum_{i=1}^p \lambda_i \widetilde{\mathbf{m}}_i \in \mathcal{N}^n$, where $0 \leq \lambda_i \leq 1$ and $\sum_{i=1}^p \lambda_i = 1$. By removing/deleting the same set of components as we did with $\mathbf{m}_1$ and $\mathbf{m}_2$ to create the sub-vectors $\widehat{\mathbf{m}}_1$ and $\widehat{\mathbf{m}}_2$, we create the sub-vector $\widehat{\mathbf{m}}_3$. It follows that $\widehat{\mathbf{m}}_3 \geq \widehat{\mathbf{m}}_1$ as $\widehat{\mathbf{m}}_2 \geq \widehat{\mathbf{m}}_1$, and each $\widetilde{\mathbf{m}}_i$ is greater than or equal to either $\mathbf{m}_1$ or $\mathbf{m}_2$.

Adding the components that were removed/deleted to form the sub-vector $\widehat{\mathbf{m}}_3$ to reconstruct that vector $\mathbf{m}_3$ will result in $\mathbf{m}_3 \geq \mathbf{m}_1$, as a consequence of the above observation. Towards this end, we partition the set $\{\widetilde{\mathbf{m}}_i\}_{i=1}^p$ into two sets $\{\widetilde{\mathbf{m}}_i^1\}_{i=1}^{p_1}$ and $\{\widetilde{\mathbf{m}}_i^2\}_{i=1}^{p_2}$, where $\forall i \in \{1,2,\ldots,p_1\}, \widetilde{\mathbf{m}}_i^1 \geq \mathbf{m}_1$, and $\forall i \in \{1,2,\ldots,p_2\}, \widetilde{\mathbf{m}}_i^2 \geq \mathbf{m}_2$. Without loss of generality, we can assume $p_1 \neq 0$. So, $\mathbf{m}_3 = (\sum_{i=1}^{p_1} \lambda_i^1 \widetilde{\mathbf{m}}_i^1) + (\sum_{i=1}^{p_2} \lambda_i^2 \widetilde{\mathbf{m}}_i^2)$, where $\sum_{i=1}^{p_1} \lambda_i^1 + \sum_{i=1}^{p_2} \lambda_i^2 = 1$, and $\forall i, j, 0 \leq \lambda_i^j \leq 1$. The above argument shows that if $j \notin \{i \in \{1,2,\ldots,n\} \mid \mid \mathbf{m}_{1_i} - \mathbf{m}_{2_i} \mid = 1\}$, then $\mathbf{m}_{3_j} \geq \mathbf{m}_{1_j}$. Suppose $j \in \{i \in \{1,2,\ldots,n\} \mid \mid \mathbf{m}_{1_i} - \mathbf{m}_{2_i} \mid = 1\}$, and $\mathbf{m}_{2_j} = \mathbf{m}_{1_j} + 1$, then $\mathbf{m}_{3_j} \geq \mathbf{m}_{1_j}$. Suppose $j \in \{i \in \{1,2,\ldots,n\} \mid \mid \mathbf{m}_{1_i} - \mathbf{m}_{2_i} \mid = 1\}$, and $\mathbf{m}_{2_j} = \mathbf{m}_{1_j} - 1$, then since $\mathbf{m}_3 \in \mathcal{N}^n$, $\mathbf{m}_{3_j} \geq \lceil \mathbf{m}_{1_j} - \sum_{i=1}^{p_2} \lambda_i^2 \rceil$. Since $p_1 \neq 0$, it follows that $\sum_{i=1}^{p_2} \lambda_i^2 < 1$, and $\mathbf{m}_{3_j} \geq \lceil \mathbf{m}_{1_j} - \sum_{i=1}^{p_2} \lambda_i^2 \rceil = \mathbf{m}_{1_j}$. □

Consider a right-closed set, $\Delta(N_1)$ with minimal elements $\{(0\ 2\ 0\ 0)^T, (1\ 0\ 0\ 1)^T\}$. The sub-vectors obtained by removing/deleting those components of these minimal elements that differ by unity yields the set $\{(2\ 0)^T, (0\ 0)^T\}$. The members of this set can be compared completely under the regular ordering. From lemma 6.1.1, we conclude that this right-closed set is convex. This is confirmed by noting that this right-closed set can be written as $P(\widehat{\mathbf{A}}, \widehat{\mathbf{b}}) \cap \mathcal{N}^4$ where

$$\widehat{\mathbf{A}} = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 \end{pmatrix} \text{ and } \widehat{\mathbf{b}} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

Continuing further, consider the set $\Delta(N_2)$ with minimal elements

$$\{(0\ 0\ 0\ 2)^T, (0\ 0\ 1\ 0), (0\ 2\ 0\ 0)^T, (1\ 0\ 0\ 1), (2\ 0\ 0\ 0)\},\ \text{where } \Delta(N_2) \subset \Delta(N_1),$$

with the lexicographic ordering of the minimal elements of $\Delta(N_2)$ is as follows

$$\underbrace{(0\ 0\ 0\ 2)^T}_{\mathbf{m}_1} <^d \underbrace{(0\ 0\ 1\ 0)^T}_{\mathbf{m}_2} <^d \underbrace{(0\ 2\ 0\ 0)^T}_{\mathbf{m}_3} <^d \underbrace{(1\ 0\ 0\ 1)^T}_{\mathbf{m}_4} <^d \underbrace{(2\ 0\ 0\ 0)^T}_{\mathbf{m}_5}.$$

From lemma 6.1.1, we note that the right-closed set defined using each pair of consecutive minimal elements define a convex set. That is, the sets $\mathcal{S}_1 = \{\mathbf{m} \in \mathcal{N}^n \mid (\mathbf{m} \geq \mathbf{m}_1) \vee (\mathbf{m} \geq \mathbf{m}_2)\}$ $\mathcal{S}_2 = \{\mathbf{m} \in \mathcal{N}^n \mid (\mathbf{m} \geq \mathbf{m}_2) \vee (\mathbf{m} \geq \mathbf{m}_3)\}$, $\mathcal{S}_3 = \{\mathbf{m} \in \mathcal{N}^n \mid (\mathbf{m} \geq \mathbf{m}_3) \vee (\mathbf{m} \geq \mathbf{m}_4)\}$, and $\mathcal{S}_4 = \{\mathbf{m} \in \mathcal{N}^n \mid (\mathbf{m} \geq \mathbf{m}_4) \vee (\mathbf{m} \geq \mathbf{m}_5)\}$ are all convex. But, $\Delta(N_2) = \bigcup_{i=1}^{4} \mathcal{S}_i$, is not convex as noted earlier.

6.2 The Invariant-Based Monitor of Figure 2.3 Revisited

Consider the PN structure N_3 shown in figure 2.2(a). The right-closed set $\Delta(N_3)$ is identified by the forty-three minimal elements in the output of the C++ program described in references [44–46] shown in figure 2.2(b). The lexicographic ordering of these forty-three minimal elements, $\{\widehat{\mathbf{m}}_i^{N_4}\}_{i=1}^{43}$ are presented in figure 6.1. Following lemma 6.1.1, each of the forty-two sets

$$\mathcal{S}_i = \{\mathbf{m} \in \mathbb{N}^n \mid (\mathbf{m} \geq \widehat{\mathbf{m}}_i^{N_3}) \vee (\mathbf{m} \geq \widehat{\mathbf{m}}_{i+1}^{N_3})\}\ i \in \{1, 2, \ldots, 42\},$$

are convex.

Additionally, $\Delta(N_3) = \bigcup_{i=1}^{42} \mathcal{S}_i$ is also convex. This follows from the fact that the $\widetilde{P}$ polytope for this example is identified by three facet-defining, right-closed inequalities:

$$
\begin{aligned}
F1 : (1\ 1\ 2\ 1\ 1\ 1\ 1)^T\mathbf{x} &\geq 3 \\
F2 : (0\ 1\ 1\ 0\ 0\ 1\ 1)^T\mathbf{x} &\geq 1, \\
F3 : (1\ 0\ 1\ 1\ 1\ 0\ 0)^T\mathbf{x} &\geq 1,
\end{aligned}
$$

where $\mathbf{x} \geq \mathbf{0}$. Replacing the inequalities with strict equalities, we get three facets of $\widetilde{P}$ where the convexity-test of the previous chapter have to be conducted to test the convexity of $\Delta(N_3)$. We will refer to these facets, with a slight abuse of notation, as $F1, F2$, and $F3$ respectively. Members of the forty-three minimal elements shown in figure 2.2(b) that lie on facets $F2$ and $F3$ belong to $\{0,1\}^7$ (i.e. they are binary integral vectors), which in turn implies that $\Upsilon(F2) \subset \{0,1\}^7$ and $\Upsilon(F3) \subset \{0,1\}^7$. Using the arguments that established corollary 1 of chapter 5, we infer that the convexity-test conducted on these facets will yield positive results. $\Upsilon(F1)$ has twenty-four members, and they belong to the set $\{0,1,2\}^7$. Additionally, there are exactly two non-zero entries in each of the twenty-four members of $\Upsilon(F1)$. It is a straightforward exercise to show that under these circumstances, the convexity-test of the previous chapter will yield positive results on facet $F1$ as well. Thus establishing the convexity of $\Delta(N_3)$ without explicit computation.

Specifically, $\Delta(N_3)$ is the set of integral vectors in the polyhedron $P(\mathbf{A}_3, \mathbf{b}_3)$ in the positive orthant described by

$$
\underbrace{\begin{pmatrix} 1 & 1 & 2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}}_{\mathbf{A}_3} \mathbf{x} \geq \underbrace{\begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}}_{\mathbf{b}_3}
$$

From theorem 3.2.10, there is an invariant-based monitor $\widehat{N}_{c,3}(\mathbf{m}_3^0, \widehat{\Theta}_3(\mathbf{m}_3^0))$ (cf. figure 2.3) that is equivalent to the minimally restrictive LESP described in figure 2.2(a). The incidence matrix for the monitor places is given by

$$\mathbf{A}_3\mathbf{C}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{pmatrix}.$$

where $\mathbf{C}_3$ is the incidence matrix of the PN structure N_3 shown in figure 2.2(a), As evidenced by the row of zeros in the monitor incidence matrix above, the constraint $\mathbf{m}_4(p_1) + \mathbf{m}_4(p_1) + 2\mathbf{m}_4(p_3) + \mathbf{m}_4(p_4) + \mathbf{m}_4(p_5) + \mathbf{m}_4(p_6) + \mathbf{m}_4(p_7) \geq 3$, which is the constraint articulated by the first row of $\mathbf{A}_4\mathbf{m}_4 \geq \mathbf{b}_4$, is invariant for any marking $\mathbf{m}_4$ reachable in $N_4(\mathbf{m}_4^0)$ for $\mathbf{m}_4^0 \in \Delta(N_3)$ (even in the absence of any supervision). There is no need to enforce this constraint with a monitor. The constraints enunciated by the second and third row of $\mathbf{A}_4\mathbf{m}_4 \geq \mathbf{b}_4$ are enforced by monitors c_1 and c_2 in figure 2.3. The initial marking of c_1 (c_2) is given by the expression $\mathbf{m}_4^0(p_2)+\mathbf{m}_4^0(p_3)+\mathbf{m}_4^0(p_6)+\mathbf{m}_4^0(p_7)-1$ ($\mathbf{m}_4^0(p_1)+\mathbf{m}_4^0(p_3)+\mathbf{m}_4^0(p_4)+\mathbf{m}_4^0(p_5)-1$). The efficacy gained when liveness can be enforced by an invariant-based monitor, when it exists, should be clear with this example.

```
Lexicographic Ordering of 43 Minimal Elements
---------------------------------------------

 1: ( 0 0 0 0 1 0 2 )
 2: ( 0 0 0 0 1 1 1 )
 3: ( 0 0 0 0 1 2 0 )
 4: ( 0 0 0 0 2 0 1 )
 5: ( 0 0 0 0 2 1 0 )
 6: ( 0 0 0 1 0 0 2 )
 7: ( 0 0 0 1 0 1 1 )
 8: ( 0 0 0 1 0 2 0 )
 9: ( 0 0 0 1 1 0 1 )
10: ( 0 0 0 1 1 1 0 )
11: ( 0 0 0 2 0 0 1 )
12: ( 0 0 0 2 0 1 0 )
13: ( 0 0 1 0 0 0 1 )
14: ( 0 0 1 0 0 1 0 )
15: ( 0 0 1 0 1 0 0 )
16: ( 0 0 1 1 0 0 0 )
17: ( 0 0 2 0 0 0 0 )
18: ( 0 1 0 0 1 0 1 )
19: ( 0 1 0 0 1 1 0 )
20: ( 0 1 0 0 2 0 0 )
21: ( 0 1 0 1 0 0 1 )
22: ( 0 1 0 1 0 1 0 )
23: ( 0 1 0 1 1 0 0 )
24: ( 0 1 0 2 0 0 0 )
25: ( 0 1 1 0 0 0 0 )
26: ( 0 2 0 0 1 0 0 )
27: ( 0 2 0 1 0 0 0 )
28: ( 1 0 0 0 0 0 2 )
29: ( 1 0 0 0 0 1 1 )
30: ( 1 0 0 0 0 2 0 )
31: ( 1 0 0 0 1 0 1 )
32: ( 1 0 0 0 1 1 0 )
33: ( 1 0 0 1 0 0 1 )
34: ( 1 0 0 1 0 1 0 )
35: ( 1 0 1 0 0 0 0 )
36: ( 1 1 0 0 0 0 1 )
37: ( 1 1 0 0 0 1 0 )
38: ( 1 1 0 0 1 0 0 )
39: ( 1 1 0 1 0 0 0 )
40: ( 1 2 0 0 0 0 0 )
41: ( 2 0 0 0 0 0 1 )
42: ( 2 0 0 0 0 1 0 )
43: ( 2 1 0 0 0 0 0 )
```

Figure 6.1: The lexicographic-ordering of the forty-three minimal elements of $\Delta(N_3)$ listed in figure 2.2(b). By lemma 6.1.1, the forty-two integral sets that are defined by the consecutive pairs in this ordering are convex, and the convex set $\Delta(N_3)$ is the union of these forty-two sets.

6.3 An illustration of Corollary 1 of Chapter 5

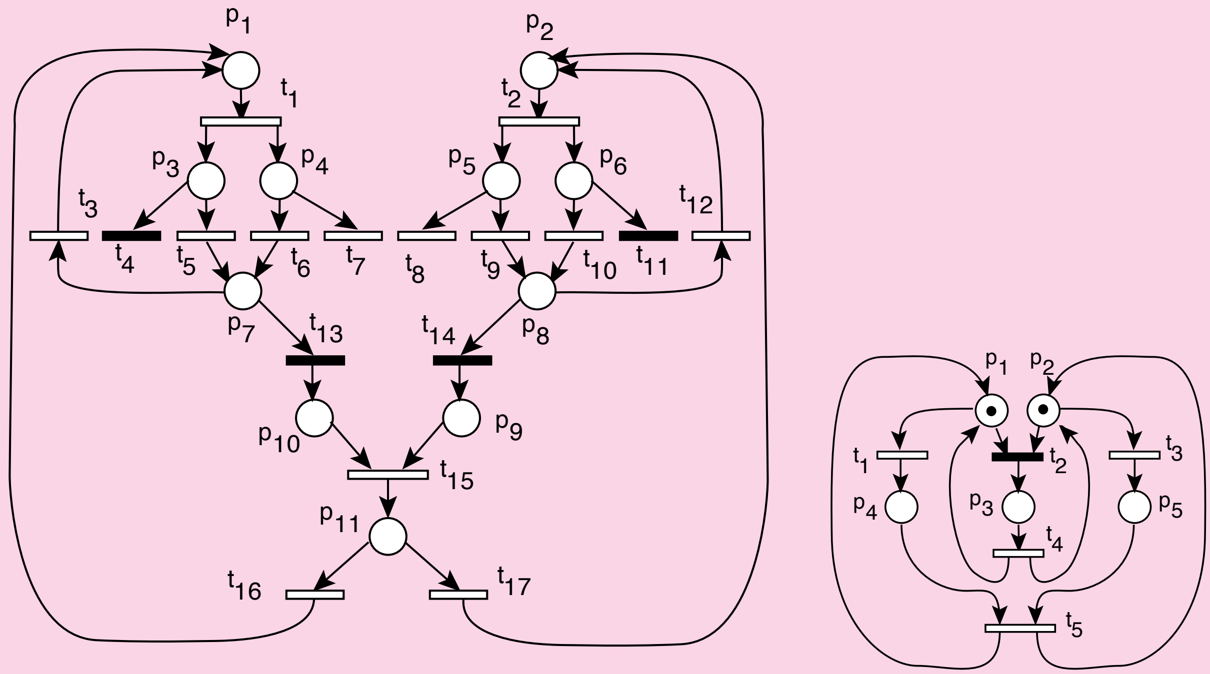
The PN structure N_7 shown in figure 6.2(a), has a $\Delta(N_7)$-set that is identified by the nine minimal elements listed below

$$(1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$$
$$(1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0)^T$$
$$(1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0)^T$$
$$(0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$$
$$(0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0)^T$$
$$(0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0)^T$$
$$(0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0)^T$$
$$(0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0)^T \quad .$$

Each of these minimal elements are members of $\{0, 1\}^{11}$. The right-closed $\Delta(N_8)$-set for the PN structure N_8 of figure rebetagamma(b) is identified by the five minimal elements listed below

$$(0\ 0\ 1\ 0\ 0)^T$$
$$(1\ 0\ 0\ 0\ 1)^T$$
$$(0\ 0\ 0\ 1\ 1)^T$$
$$(0\ 1\ 0\ 1\ 0)^T$$
$$(1\ 1\ 0\ 0\ 0)^T$$

which belong to the set $\{0,1\}^5$. From corollary 1 of chapter 5, we conclude that both these sets are convex.



(a) N_7 (b) N_8

Figure 6.2: Two PN structures with right-closed $\Delta(N)$-sets.

Chapter 7

Future Work

In this section, we discuss some of the results that can be explored as a future research direction. In previous chapter, we presented the lemma 6.1.1, where it presents a sufficient condition under which, the union of two right-closed sets is convex. There are necessary and sufficient conditions for a finite union of a set of convex polytopes to be convex [43, 47]. However, these results do not translate to necessary and sufficient conditions for a finite union of right-closed sets of integral-points to be convex. Since every right-closed set is a finite union of a collection of right-closed sets with just one minimal element, even computationally tractable sufficient conditions for the convexity of right-closed sets can have an impact on the development of algorithms for the synthesis of invariant-based monitors for liveness enforcement.

Also, as a future direction of research, we note that there is room to improve the efficiency of the algorithm. Convexity tests that do not rely on the computation of the convex hull could possibly yield faster algorithms. On the LP-based algorithm, there is still room to improve the efficiency of the algorithm. The way that the set $\mathcal{L}(\mathcal{F}_i)$ is constructed can be revised in such a manner that the cardinality of the set is reduced. There is even a possibility that this construction can be done in a way that the feasibility LP can be eliminated from the algorithm, which would result in a much faster algorithm. If a probabilistic solution to testing convexity is satisfactory, randomized algorithms for convexity testing can also be explored.

In the context of LESPs for PN models, there can be monitors that enforce liveness that are *not* invariant-based. Figure 7.1(b) presents a monitor $\widehat{N}_{9,c}(\mathbf{m}_9^0, \widehat{\Theta}_9(\mathbf{m}_9^0))$, where $\widehat{\Theta}_9(\mathbf{m}_9^0) = \mathbf{m}_9^0(p_1) + \mathbf{m}_9^0(p_2) - 3$, that enforces liveness in the general FCPN $N_9(\mathbf{m}_9^0)$ of figure 7.1(a) for any $\mathbf{m}_9^0 \in \Delta(N_9)$. The incidence matrix for the monitor place c is given by the row vector $(1\ 0\ -3)$, which is not in the row-space of the incidence matrix C_9 of the PN structure N_7. Consequently, the liveness enforcing monitor of figure

7.1(b) is not an invariant-based monitor. The set $\Delta(N_9)$ is characterized by the minimal elements $\{(2\ 0)^T, (0\ 3)^T\}$, is not convex (cf. the discussion that accompanies figure 4.3(a) and the right-closed set $\mathcal{M}_1$).

For a specific initial marking $\widehat{\mathbf{m}}_9^0 = (2\ 0)^T (\in \Delta(N_9))$, we have $\widehat{\Theta}_9(\widehat{\mathbf{m}}_9^0) = -1$, which is represented by an unfilled token in p_2 in figure 7.1(b). As the token loads of monitor places serve a book-keeping purpose, we can permit a negative token load in monitor places as long as transition t_3 is enabled in $\widehat{N}_{9,c}(\mathbf{m}_9^0, \widehat{\Theta}_9(\mathbf{m}_9^0))$ only when there are at least three tokens in c. It can be shown that for any $\mathbf{m}_9^0 \in \Delta(N_9)$, none of the sub-markings in $\mathcal{N}^2 - \Delta(N_9)$ are potentially reachable in $N_{9c}(\mathbf{m}_9^0, \Theta(\mathbf{m}_9^0)$, which means the $\widehat{N}_{9,c}(\mathbf{m}_9^0, \widehat{\Theta}_9(\mathbf{m}_9^0))$ is live. The control action affected by the monitor c implicitly defines an LESP for $N_9(\mathbf{m}_9^0)$.

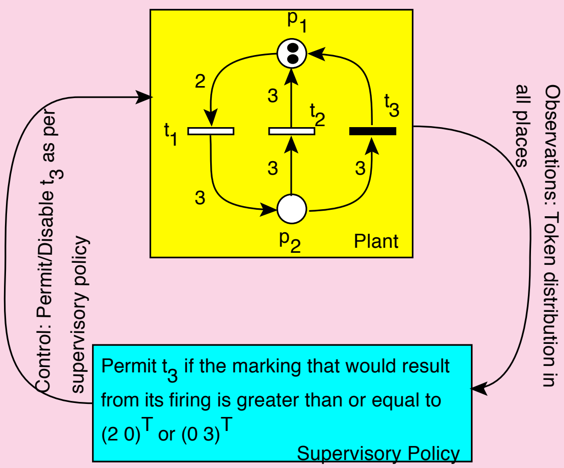
If we initialized the PN structure N_9 with the marking $\widehat{\mathbf{m}}_9^0 = (1\ 3)^T$ ($\Rightarrow \Theta(\widehat{\mathbf{m}}_9^0) = 1$), transition t_3 would be permitted to fire by the minimally restrictive LESP of figure 7.1(a), while the same transition would not be permitted to fire in $\widehat{N}_{9,c}(\mathbf{m}_9^0, \widehat{\Theta}_9(\mathbf{m}_9^0))$ of figure 7.1(b). That is, the LESP implicitly defined by the action of the monitor place in figure 7.1(b) is not minimally restrictive. Investigating the parallel of theorem 3.2.10 for monitors that enforce liveness that are not invariant-based, where we permit the presence of negative tokens in monitors, is suggested as a future research topic.

Reference [48] notes that the minimally restrictive LESP for a PN $N(\mathbf{m}^0)$ with a right-closed $\Delta(N)$-set can be implemented using a *Disjunctive Normal Form* (DNF) formula associated with each controllable transition of N. This is best illustrated by example.

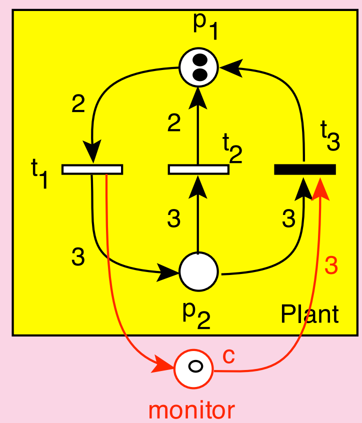
Consider the PN structure N_9 of figure 7.1, operating under the supervision of a policy that permits the firing of the controllable transition t_3 at a marking $\mathbf{m}$ if and only if the DNF

$$\Theta_{t_3}(\mathbf{m}) = (\mathbf{m}(p_1) \geq 1) \vee (\mathbf{m}(p_2) \geq 3).$$

is true. This supervisory policy is equivalent to the minimally restrictive LESP of figure 7.1(a). For illustrative purposes, this policy is shown in figure 7.2(a). Since $\Delta(N_9)$ ($= \mathcal{M}_1$ (cf. section 4.3)) is not convex, there can be no invariant-based monitor that is equivalent to the minimally restrictive LESP



(a) The minimally restrictive LESP



(b) Alternate Monitor Placement

Figure 7.1: (a) A minimally-restrictive LESP that ensures all reachable markings of the plant FCPN PN $N_9(\mathbf{m}_9^0)$ are in the right-closed set $\Delta(N_9)$ identified by the minimal elements $\{(2\ 0)^T, (0\ 3)^T\}$. This policy is a minimally restrictive LESP for any $\mathbf{m}_7^0 \in \Delta(N_7)$. There is no LESP for $N_9(\mathbf{m}_9^0)$ if $\mathbf{m}_9^0 \notin \Delta(N_9)$. (b) An illustration of liveness enforcement for $N_9(\mathbf{m}_9^0)$ of figure (a) using a monitor $\widehat{N}_{9,c}(\mathbf{m}_9^0, \widehat{\Theta}_9(\mathbf{m}_9^0))$ that is *not* invariant-based, where $\mathbf{m}_9^0 \in \Delta(N_9)$. The incidence matrix for the monitor is $(1\ 0\ -3)$ which is not in the row-space of the incidence matrix of the PN structure N_9. The unfilled circle in the monitor c represents a token load of -1. However, transition t_3 is enabled if and only if there are at least three tokens in p_2 and c.

of figure 7.1(a), which is equivalent to the LESP of figure 7.2(a).

There is theoretical support to show that the set of markings that make DNFs like $\Theta_{t_3}(\mathbf{m})$ true is right-closed. In some cases, this right-closed set may be convex. For instance, the set of markings that make $\Theta_{t_3}(\mathbf{m})$ true, satisfy the inequality

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{m} \geq \begin{pmatrix} 1 \\ 6 \end{pmatrix}.$$

Therefore, this set is convex.

Inspired by the results of this thesis, derived in the context of invariant-based monitors that are equivalent to $\Delta(N)$-set based LESPs, we infer that there should be a construction where the presence of a non-zero token loads in monitor places implies membership in the set of markings that makes $\Theta_{t_3}(\mathbf{m})$ true. This requires us to enhance the standard PN-semantics with the addition of *record-keeping places* whose token loads can be negative. This is in addition to the regular PN-semantics that requires each input place of a transition to have sufficient tokens, *ex ante*, to permit its firing, which is accompanied by an appropriate change in the token loads, *ex post* the firing of the transition. The record-keeping places do not prevent firings of transitions due to insufficient tokens, as with regular places. This is illustrated in figure 7.2(b).

The incidence matrix for the record-keeping monitor c_1 is given by

$$\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} -2 & 3 & 1 \\ 3 & -3 & -3 \end{pmatrix} = \begin{pmatrix} -2 & 3 & 1 \end{pmatrix},$$

while that for the record-keeping monitor c_2 is given by

$$\begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} -2 & 3 & 1 \\ 3 & -3 & -3 \end{pmatrix} = \begin{pmatrix} 3 & -3 & -3 \end{pmatrix}.$$

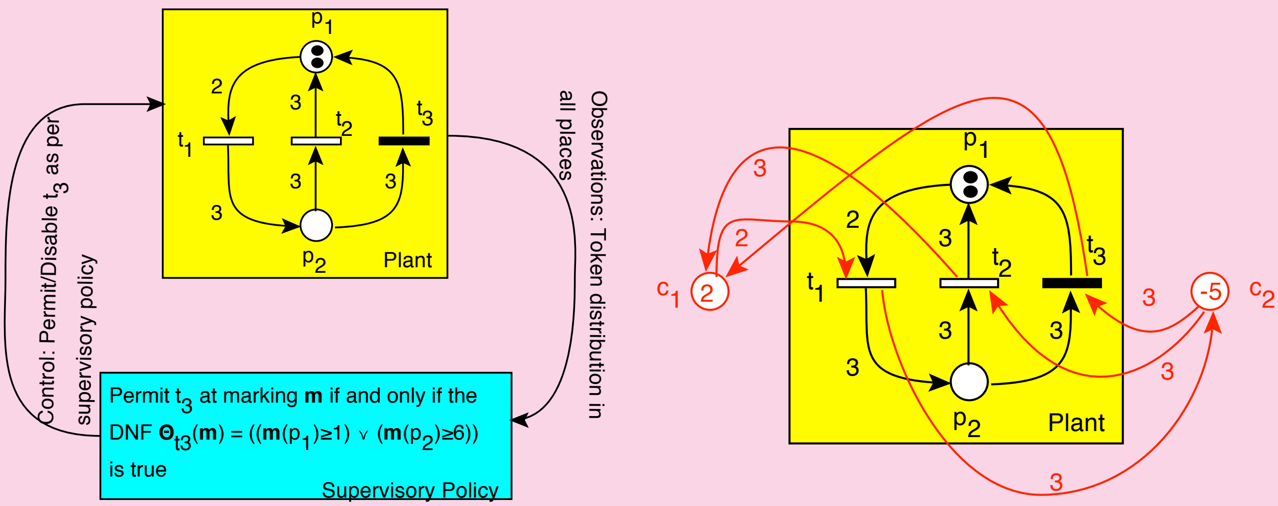
This is appropriately shown by the arcs and arc-weights associated with c_1 and c_2 in figure 7.2(b). The initial marking of c_1 is determined by the expression $\mathbf{m}^0(c_1) = \mathbf{m}^0(p_1) - 1 + 1 = \mathbf{m}^0(p_1)$, and the initial marking of c_2 is given by the expression $\mathbf{m}^0(c_2) = \mathbf{m}^0(p_2) - 6 + 1 = \mathbf{m}^0(p_2) - 5$. For the initial marking $(2\ 0)^T$, we obtain $\mathbf{m}^0(c_1) = 1$ and $\mathbf{m}^0(c_2) = -5$ (cf. figure 7.2(b)). Let us suppose we have a policy that permits t_3 if and only if c_1 or c_2 has a positive token load.

Consider the following firing string under the influence of the aforementioned policy:

$$\underbrace{\begin{pmatrix} 2 \\ 0 \\ 2 \\ -5 \end{pmatrix}}_{\text{permit } t_3:Yes} \overset{t_1}{\rightarrow} \underbrace{\begin{pmatrix} 0 \\ 3 \\ 0 \\ -2 \end{pmatrix}}_{\text{permit } t_3:No} \overset{t_2}{\rightarrow} \underbrace{\begin{pmatrix} 3 \\ 0 \\ 3 \\ -5 \end{pmatrix}}_{\text{permit } t_3:Yes} \overset{t_1}{\rightarrow} \underbrace{\begin{pmatrix} 1 \\ 3 \\ 1 \\ -2 \end{pmatrix}}_{\text{permit } t_3:Yes} \overset{t_3}{\rightarrow} \underbrace{\begin{pmatrix} 2 \\ 0 \\ 2 \\ -5 \end{pmatrix}}_{\text{permit } t_3:Yes}$$

The control action prescribed by the aforementioned policy is placed below each marking. The controllable transition t_3 is permitted if c_1 or c_2 has a positive token load. It is not hard to see that this policy is equivalent to the minimally restrictive LESP of figures 7.1(a) and 7.2(a). The monitors c_1 and c_2 are record-keeping places, whose token loads can be negative. Additionally, the fact that there is an insufficient number of tokens in any of them does not prevent the firing of transitions – this should be apparent in the firing string presented above. We refer to this construction as a *Generalized* monitor construction. There are PNs (like N_9 as shown here) where there is no invariant-based monitor that is equivalent to the minimally restrictive LESP, but there is a generalized monitor construction that yields a LESP that is minimally restrictive. We suggest explorations into this class of problems as a future research topic, as well.

The computational examples in this thesis utilized the various functional primitives in the software package *Polymake* [33,34], and the mathematical programming components were executed using *lp_solve* [49]. Results from each of these computational tools were integrated manually, and it was very time consuming. Another direction for future work could be the seamless integration of the polyhedral computation with the mathematical programming concepts developed in this thesis in a manner that is transparent to the user. The convexity testing procedures of chapter 5 are inherently amenable to



(a) DNF-based minimally restrictive LESP

(b) *Generalized* Monitor Placement

Figure 7.2: (a) A DNF-based minimally restrictive LESP for the PN $N_9(\mathbf{m}_9^0)$ of figure 7.1 obtained using the methods of reference [48]. (b) A *generalized* monitor construction that is equivalent to the minimally restrictive LESP for $N_9(\mathbf{m}_9^0)$. The monitors c_1 and c_2 are *record-keeping* places, whose token-loads can be negative in course of transition firings. That is, the lack of sufficient tokens in these special-places does not prevent the firing of transitions. The supervisory policy permits t_3 if and only if one of the monitors has a positive token load. Based on the results in thesis, there is no invariant-based monitor that is equivalent to his generalized monitor.

GPU-accelerated computing. Explorations into the implementation-side of the work presented in this thesis will be valuable.

The literature on supervisory control of PNs contains explorations into the fault-tolerant implementation of supervisory policies, that are meant to combat sensor failures. A sensor failure could result in incorrect information about the tokens in places, which could lead to incorrect supervision – and The literature on supervisory control of PNs contains explorations into the fault tolerant implementation of supervisory policies, that are meant to combat sensor failures. A sensor failure could result in incorrect information about the tokens in places, which could lead to incorrect supervision- and eventually to a livelocked-state. The methods used in references [50, 51] rely on coding-theoretic techniques that involve the replication of information in a manner such that incorrect information can be be corrected by carefully placed redundancies within the system. Just as with monitor place constructions, this scheme for fault-tolerance involves enhancements to the original PN structure with extra places and arcs to- and from- existing transitions. There are necessary and sufficient conditions for the existence of a fault-tolerant version of a supervisory for a given PN model of a DEDS system that parallels the convexity condition for the existence of a minimally restrictive monitor construction for liveness. A fruitful direction for future research could include investigations into paralleling the results in this thesis in the domain of fault-tolerant implementations of supervisory policies for DEDS systems modeled as PNs. corrected by carefully placed redundancies within the system. Just as with monitor place constructions, this scheme for fault-tolerance involves enhancements to the original PN structure with extra places and arcs to- and from- existing transitions. There are necessary and sufficient conditions for the existence of a fault-tolerant version of a supervisory for a given PN model of a DEDS system that parallels the convexity condition for the existence of a minimally restrictive monitor construction for liveness. A fruitful direction for future research could include investigations into paralleling the results in this thesis in the domain of fault-tolerant implementations of supervisory policies for DEDS systems modeled as PNs.

Chapter 8

Conclusions

Throughout this dissertation, we have focused on a family of PN structures where for any instance N, the existence of a *liveness enforcing supervisory policy* (LESP) for $N(\mathbf{m}^0)$ is sufficient to infer that there is an LESP for $N(\widehat{\mathbf{m}}^0)$, where $\widehat{\mathbf{m}}^0 \geq \mathbf{m}^0$. Consequently, the set of initial markings for which there is an LESP, $\Delta(N)$, is *right-closed*, and is characterized by its finite set of minimal elements $\min(\Delta(N)$. Additionally, $\Delta(N)$ is control-invariant with respect to N, that is, the firing of any uncontrollable transition at $\mathbf{m}^i \in \Delta(N)$ will result in a new marking that is also in $\Delta(N)$. For $\mathbf{m}^0 \in \Delta(N)$, the minimally restrictive LESP for $N(\mathbf{m}^0)$ disables the firing of any controllable transition at a reachable marking, if its firing would result in a new marking that is not in $\Delta(N)$. We showed that for these type of PNs, the necessary and sufficient condition to have equivalent minimally restrictive LESP and invariant-based monitor is the integer-convexity of the $\Delta(N)$. Additionally for those cases where there is no invariant-based monitor that is equivalent to the minimally restrictive LESP, we presented a synthesis procedure for a more restrictive invariant-based monitor that implicitly defines an LESP for $N(\mathbf{m}^0)$.

As an algorithmic approach to test this necessary and sufficient condition, we showed that a right-closed set $\mathcal{M}$ is integer-convex if and only if the condition

$$\min(\mathcal{M}) = \min(Int(conv(\mathcal{M})))$$

is satisfied. Given the fact that computing the $\min(Int(conv(\mathcal{M})))$ can be computationally an infeasible procedure, we presented an alternative condition for verifying the convexity. We suggested that using the convex hull of the minimal elements of $\mathcal{M}$ along with their elevated counterparts can help us to present a more feasible condition.

We defined new sets $\mathcal{V}$ and $\widetilde{P}$ as subsets of the original set $\mathcal{M}$. These two sets share all the right-closed facets, defining the $conv(\mathcal{M})$. By utilizing this fact, we presented a new criteria for verifying the convexity of a right-closed set $\mathcal{M}$:

$$\forall \widetilde{\mathbf{x}} \in conv(\Upsilon(\mathcal{F}_i)), \lceil \widetilde{\mathbf{x}} \rceil \in \mathcal{M})$$

Based on this observation we presented the LP-based algorithm for testing the convexity of a right-closed set, which for a fixed dimension n, can verify the convexity of such a set in a polynomial time. We then proposed a gird-search based algorithm, for testing the convexity of integral right-closed set, which although not a polynomial time procedure, is a method that verifies the convexity in a reasonable time complexity. The complexity analysis of this algorithm showed that the number of lattice points inside a polytope does not increase in a exponential manner when its volume increases by an exponential factor.

We concluded this thesis by presenting possible approaches to improve the running time of convexity testing by developing sufficient conditions for integer-convexity. The violation of these conditions, which can be tested easily, will rule out convexity without additional computation.

We suggest explorations into probabilistic convexity testing as another approach to mitigating the inevitable computational issues that we would have to contend with for high-dimensional polytopes. The development of a family of probabilistic algorithms with a high probability of detection of convexity/non-convexity would be highly desirable.

References

[1] R. Valk and M. Jantzen, "The residue of vector sets with applications to decidability problems in Petri nets," *Acta Informatica*, vol. 21, pp. 643–674, 1985.

[2] B. Alpern and F. B. Schneider, "Defining liveness," *Information Processing Letters*, vol. 21, no. 4, 1985.

[3] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM journal on control and optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[4] W. Wonham and P. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM Journal of Conrol and Optimization*, vol. 25, no. 3, pp. 637–659, May 1987.

[5] R. Sreenivas, "Towards a system theory for interconnected condition/event systems," Ph.D. dissertation, Carnegie Mellon University, 1990.

[6] R. Sreenivas, "An application of independent, increasing, free-choice petri nets to the synthesis of policies that enforce liveness in arbitrary petri nets," *Automatica*, vol. 34, no. 12, pp. 1613–1615, December 1998.

[7] R. Sreenivas, "On supervisory policies that enforce liveness in in a class of completely controlled petri nets obtained via refinement," *IEEE Transactions on Automatic Control*, vol. 44, no. 1, pp. 173–177, January 1999.

[8] R. Sreenivas, "On supervisory policies that enforce liveness in completely controlled petri nets with directed cut-places and cut-transitions," *IEEE Transactions on Automatic Control*, vol. 44, no. 6, pp. 1221–1225, June 1999.

[9] R. Sreenivas and B. Krogh, "On condition/event systems with discrete state realizations," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 1, pp. 209–236, 1991.

[10] R. Sreenivas, "A note on deciding the controllability of a language K with respect to a language L," *IEEE Trans. on Automatic Control*, vol. 38, no. 4, April 1993.

[11] R. Sreenivas, "On a weaker notion of controllability of a language K with respect to a language L," *IEEE Trans. on Automatic Control*, vol. 38, no. 9, September 1993.

[12] R. Sreenivas, "On minimal representations of petri net languages," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 799–804, May 2006.

[13] P. Ramadge and W. Wonham, "The control of Discrete Event Systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, January 1989.

[14] P. Ramadge and W. Wonham, "Modular feedback logic for discrete event systems," *SIAM J. Control and Optimization*, vol. 25, no. 5, pp. 1202–1218, September 1987.

[15] J. L. Peterson, *Petri net theory and the modeling of systems.* Prentice Hall PTR, 1981.

[16] C. Reutenauer, *The mathematics of Petri nets.* Prentice-Hall, Inc., 1990.

[17] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[18] M. H. T. Hack, "Analysis of production schemata by petri nets," DTIC Document, Tech. Rep., 1972.

[19] J. Desel and J. Esparza, *Free choice Petri nets.* Cambridge university press, 2005, vol. 40.

[20] R. Sreenivas, "On the existence of supervisory policies that enforce liveness in partially controlled free-choice petri nets," *Automatic Control, IEEE Transactions on*, vol. 57, no. 2, pp. 435–449, 2012.

[21] L. Rademacher and S. Vempala, "Testing geometric convexity," in *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science.* Springer, 2005, pp. 469–480.

[22] E. Salimi, N. Somnath, and R. Sreenivas, "A tutorial on the synthesis of the maximally permissive liveness enforcing supervisory policy in discrete-event/discrete-state systems modeled by a class of general petri nets," in *Proceedings of The First Indian Control Conference (ICC 2015)*, M. Vidyasagar, Ed., Indian Institute of Technology, Madras, January 2015.

[23] E. Salimi, N. Somnath, and R. Sreenivas, "A Software Tool for Live-Lock Avoidance in Systems Modeled Using a Class of Petri Nets," *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, vol. 5, no. 2, pp. 1–13, April 2015.

[24] E. Salimi, N. Somnath, and R. Sreenivas, "On supervisory policies that enforce liveness in controlled petri nets that are similar," in *Proceedings of the 7th IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and the 7th IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, Angkor Wat, Cambodia, July 2015.

[25] E. Salimi and R. Sreenivas, "On invariant-based monitors that enforce liveness in a class of partially controlled general petri nets," *IEEE Transactions on Automatic Control*, 2015, to appear.

[26] R. Sreenivas, "On asymptotically efficient solutions for a class of supervisory control problems," *IEEE Transactions on Automatic Control*, vol. 41, no. 12, pp. 1736–1750, December 1996.

[27] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive petri net controller using the theory of regions," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 1, pp. 137–141, 2003.

[28] A. Giua, "Petri nets as discrete event models for supervisory control," Ph.D. dissertation, Rensselaer Polytechnic Institute, 1992.

[29] J. Moody and P. J. Antsaklis, *Supervisory control of discrete event systems using Petri nets*. Springer, 1998, vol. 8.

[30] S. Chandrasekaran, N. Somnath, and R. Sreenivas, "A software tool for the automatic synthesis of minimally restrictive liveness enforcing supervisory policies for a class of general petri net models of manufacturing-and service-systems," *Journal of Intelligent Manufacturing*, pp. 1–14, 2014.

[31] K. Fukuda et al., "Frequently asked questions in polyhedral computation," Report http://www.inf.ethz.ch/personal/fukudak/polyfaq/polyfaq.html, ETH, Zürich, accessed: 2015-6-1.

[32] D. M. Mount, "Geometric intersection," in *Handbook of Discrete and Computational Geometry, chapter 33*. Citeseer, 1997.

[33] E. Gawrilow and M. Joswig, *Polymake: a framework for analyzing convex polytopes*. Springer Basel AG, 2000, pp. 43–73.

[34] "Polymake.org," http://polymake.org/doku.php, accessed: 2013-08-7.

[35] B. Chazelle, "An optimal convex hull algorithm in any fixed dimension," *Discrete & Computational Geometry*, vol. 10, no. 1, pp. 377–409, 1993.

[36] M. Beck and S. Robins, *Computing the continuous discretely: Integer-point enumeration in polyhedra*. Springer, 2007.

[37] H. S. Wilf, *generatingfunctionology*. Elsevier, 2013.

[38] R. Sreenivas, "On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modeled by controlled Petri nets," *IEEE Transactions on Automatic Control*, vol. 42, no. 7, pp. 928–945, July 1997.

[39] N. Somnath and R. Sreenivas, "On deciding the existence of a liveness enforcing supervisory policy in a class of partially controlled general free-choice petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 4, pp. 1157–1160, 2013.

[40] R. Sreenivas, "On a decidable class of partially controlled petri nets with liveness enforcing supervisory policies," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 5, pp. 1256–1261, August 2013.

[41] R. S. Sreenivas, "Some observations on supervisory policies that enforce liveness in partially controlled free-choice petri nets," *Mathematics and Computers in Simulation*, vol. 70, no. 5, pp. 266–274, 2006.

[42] J. A. De Loera, "The many aspects of counting lattice points in polytopes," *Mathematische Semesterberichte*, vol. 52, no. 2, pp. 175–195, 2005.

[43] A. Bemporad, K. Fukuda, and F. D. Torrisi, "Convexity recognition of the union of polyhedra," *Computational Geometry*, vol. 18, no. 3, pp. 141–154, 2001.

[44] S. Chandrasekaran and R. Sreenivas, "On the automatic generation of the minimally restrictive liveness enforcing supervisory policy for manufacturing- and service-systems modeled by a class of general free choice petri nets," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC-13)*, Paris, France, April 2013, session WeC01.3.

[45] S. Chandrasekaran, "Object-oriented implementation of the minimally restrictive liveness enforcing supervisory policy in a class of petri nets," M.S. thesis, University of Illinois at Urbana-Champaign, Industrial and Enterprise Systems Engineering, December 2012.

[46] S. Chandrasekaran and R. Sreenivas, "A software tool for the synthesis of supervisory policies that avoid livelocks in petri net models of manufacturing- and service-systems," in *Proceedings of the XVI Annual International Conference of the Society of Operations Management (SOM-12)*, New Delhi, India, December 2012.

[47] I. Bárány and K. Fukuda, "A case when the union of polytopes is convex," *Linear Algebra and its Applications*, vol. 397, pp. 381–388, 2005.

[48] V. Deverakonda and R. Sreenivas, "On a sufficient information structure for supervisory policies that enforce liveness in a class of general petri nets," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1915–1921, July 2015.

[49] "lp_solve 5.5.2.0," http://lpsolve.sourceforge.net/5.5/, accessed: 2015-06-2.

[50] L. Li, C. Hadjicostis, and R. S. Sreenivas, "Designs of bisimilar petri net controllers with fault tolerance capabilities," *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, vol. 38, no. 1, pp. 207–217, January 2008.

[51] L. Li, C. Hadjicostis, and R. S. Sreenivas, "Fault detection and identification in petri net controllers," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC)*, Bahamas, December 2004, pp. 5248–5253.