Optimal switching sequence for switched linear systems

Zeyang Wu* and Qie He†

Department of Industrial & Systems Engineering, University of Minnesota, USA

July 1, 2018

Abstract

We study the following optimization problem over a dynamical system that consists of several linear subsystems: Given a finite set of $n \times n$ matrices and an $n$-dimensional vector, find a sequence of $K$ matrices, each chosen from the given set of matrices, to maximize a convex function over the product of the $K$ matrices and the given vector. This simple problem has many applications in operations research and control, yet a moderate-sized instance is challenging to solve to optimality for state-of-the-art optimization software. We propose a simple exact algorithm for this problem. Our algorithm runs in polynomial time when the given set of matrices has the oligo-vertex property, a concept we introduce in this paper for a set of matrices. We derive several sufficient conditions for a set of matrices to have the oligo-vertex property. Numerical results demonstrate the clear advantage of our algorithm in solving large-sized instances of the problem over one state-of-the-art global solver. We also pose several open questions on the oligo-vertex property and discuss its potential connection with the finiteness property of a set of matrices, which may be of independent interest.

1 Introduction

Many real-world systems exhibit significantly different dynamics under various modes or conditions, for example a manual transmission car operating at different gears, a chemical reactor under different temperatures and flow rates of reactants, and a group of cancer cells responding to different drugs. Such phenomena can be modeled under a unified framework of switched systems. A switched system is a dynamical system that consists of several subsystems and a rule that specifies the switching among the subsystems. Finding a switching rule to optimize the dynamics of a switched system under certain criteria has found numerous applications in power system operations, chemical process control, air traffic management, and medical treatment design [36, 24, 23, 15]. In this paper, we study the following discrete-time switched linear system:

$$x(k + 1) = T_k x(k), \quad T_k \in \Sigma, \; k = 0, 1, \ldots ,$$

where $x(k)$ is an $n$-dimensional real vector that captures the system state at period $k$, the set $\Sigma$ contains $m$ given $n \times n$ real matrices, each of which describes the dynamics of a linear subsystem, and the initial vector $x(0)$ is a given $n$-dimensional real vector $a$. Such a system with switching only at fixed time instants appear in many practical applications, and is also employed to approximate

---

*Email address: wuxx1164@umn.edu.
†Email address: qhe@umn.edu. Corresponding author.
the more complex dynamics of a continuous-time hybrid system with switching rules defined over the real line [36, 23].

We are interested in the following optimization problem (P) related to the system in [1]

Given a switched linear system described in [1], a positive integer $K$, and a convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, find a sequence of $K$ matrices $T_0, T_1, \ldots, T_{K-1} \in \Sigma$ to maximize $f(x(K))$.

One type of such convex functions are the $\ell_p$ norms.

**Example 1.** Consider a switched linear system consisting of two subsystems with system matrices $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, an initial vector $a = (2, 1)^{\top}$, and $K = 8$. Figure 1 illustrates the trajectory of $x(k)$ under three switching sequences, with the final state $x(8)$ being $(53, 23)^{\top}$, $(58, 41)^{\top}$, and $(71, 41)^{\top}$, respectively.

![Figure 1: The trajectory of $x(k)$ under different matrix sequences](image)

We give three examples below to illustrate the applications of Problem (P) and its connection to other problems in control and optimization.

The first example is on design of treatment plans. Antibiotic resistance renders diseases that were once easily treatable dangerous infections, and has become one of the most pressing public health problems around the world. Several groups of researchers studied how to design sequential antibiotic treatment plans to restore susceptibility after bacteria develop resistance [26, 28]. They model the percentages of $n$ genotypes of an enzyme produced by bacteria in a population with vector $x(k)$ after $k$ periods of treatment, and a probability transition matrix to model the mutation rates among $n$ genotypes under each antibiotic. The goal is to design a sequence of antibiotics to maximize the percentage of the wild type at the end of the treatment, which is sensitive to all antibiotics. The treatment design problem is equivalent to solve (P) with $a = e_1$, a unit vector with the first component being 1 which denotes 100% wild type in the beginning, and $f(x(K)) = -e_1^{\top} x(K)$. In the same vein, (P) can model the sequential therapy design problem for many other diseases when $x(k)$ describes related biometrics of a patient at period $k$ and each matrix models the evolution of patient biometrics under a particular treatment [15].

The second example is the matrix mortality problem in control [6, 7]. Given a positive integer $k$, a set of matrices is said to be $k$-mortal if the zero matrix can be expressed as a product of $k$ matrices in the set (duplication allowed). A set of matrices is said to be mortal if it is $k$-mortal for
some finite $k$. The matrix mortality problem captures the stability of switched linear systems under certain switching rules. It can be shown that a finite set of $n \times n$ non-negative matrices is $k$-mortal if and only if the optimal objective value of (P) is 0 with $a = 1$, $K = k$, and $f(x(K)) = -1^\top x(K)$, where $1$ is a $n$-dimensional vector with each component being 1.

The third example concerns the joint spectral radius of a set of matrices, an important quantity which has found many applications in wavelet functions, constrained coding, and network security management, etc [17]. The joint spectral radius of a finite set $\Sigma$ of matrices [33] is defined as $\rho(\Sigma) = \limsup_{k \to \infty} \hat{\rho}_k(\Sigma, \| \cdot \|)$, where

$$\hat{\rho}_k(\Sigma, \| \cdot \|) = \max \{\|T_{k-1}T_{k-2}\cdots T_0\|^{1/k} \mid T_j \in \Sigma, j = 0, \ldots, k - 1\}$$

(2)

and $\| \cdot \|$ is some matrix norm. If we select the matrix norm in (2) to be induced by the $\ell_p$ norm of a vector, then

$$(\hat{\rho}_k(\Sigma, \| \cdot \|))^k = \sup_{\|a\|_p = 1} \max \{\|x(K)\|_p \mid \{1\}\}.$$  

(3)

Observe that the inner optimization problem of the right-hand side of (3) is a special case of (P) with the convex function $f(x) = \|x\|_p$. In general, let $v^*$ be the optimal objective value of (P) with $f(x) = \|x\|$ for some norm $\| \cdot \|$ and an initial vector $a$. Then $(v^*)^{1/k}$ provides a lower bound of the quantity $\hat{\rho}_k(\Sigma, \| \cdot \|)$.

A simple way to solve (P) is to enumerate all possible matrix sequences, but such an approach quickly becomes impractical as $m$ and $K$ increase. Even for $m = 5$ and $K = 30$, we need to enumerate $5^{30}$ solutions, which is unrealistic in practice. Another general approach to solve (P) is to formulate it as a mixed-integer nonlinear optimization problem, which can be solved by global optimization solvers, but the problem size that can be handled by state-of-the-art commercial solvers is also limited. In addition, the time complexity of the tree-based search algorithms employed by these global solvers is difficult to analyze in general. In many applications, problem (P) has to be solved repeatedly with different parameters, so it is of vital importance to have a fast algorithm for (P).

We now introduce our results. We develop a simple dynamic programming algorithm to solve (P) exactly, where several linear programs are solved at each iteration. One advantage of our algorithm is that it does not require any additional property of $f$ such as smoothness or strong convexity. Our algorithm is very efficient in practice, based on computational results in Section 6. To analyze the time complexity of the algorithm, we assume that all input data are integers and the value of the convex function $f$ can be queried through an oracle in constant time; we adopt the random-access machine [29] as the model of computation, in which each basic operation (addition, comparison, multiplication, etc.) is assumed to take the same amount of time and the time complexity of an algorithm is the number of steps/operations required to execute the algorithm. We define the following notations that are useful for presenting the time-complexity results. Given a finite set $\Sigma$ of $n \times n$ real matrices and a vector $a \in \mathbb{R}^n$, let

$$P_k(\Sigma, a) := \text{conv}\{x(k) \mid x(k) = T_{k-1}\cdots T_0a, T_j \in \Sigma, j = 0, \ldots, k - 1\}$$

be the convex hull of all possible values of $x(k)$ in (1) for each integer $k \geq 0$. Let $N_k(\Sigma, a)$ be the number of extreme points of $P_k(\Sigma, a)$ and

$$N_k(\Sigma) = \sup_{a \in \mathbb{R}^n} \{N_k(\Sigma, a)\}.$$  

We introduce the following concept for a set of matrices.
**Definition 1.** A set of matrices \( \Sigma \) is said to have the oligo-vertex property if there exists \( \alpha > 0 \), positive integer \( k_0 \), and positive constant \( d \) such that \( N_k(\Sigma) \leq \alpha k^d \) for any \( k \geq k_0 \).

The oligo-vertex property of a set of matrices indicates the number of extreme points of \( P_k(\Sigma, a) \) grows at most polynomially in \( k \) for any initial vector \( a \), despite the number of possible values of \( x(k) \) grows exponentially with \( k \) in general. With the Big O notation commonly used in computer science, the oligo-vertex property basically states that \( N_k(\Sigma) = O(k^d) \) as \( k \to \infty \) for some positive constant \( d \).

**Our contributions**

We summarize the contributions of this paper as follows.

1. We present a simple dynamic programming algorithm to solve (P) exactly. Our algorithm does not require any additional property of \( f \) other than convexity. The running time of our algorithm is \( O(m^2 n^{1.5} (\log n + \log M) \sum_{k=0}^{K-1} k N_k(\Sigma)^2) \), and can be reduced to \( O(m \log m \sum_{k=0}^{K-1} N_k(\Sigma) + m \sum_{k=0}^{K-1} N_k(\Sigma) \log N_k(\Sigma)) \) when \( n = 2 \), where \( M \) is the maximum absolute value of the entries of \( A_1, \ldots, A_m \), and \( a \).

2. We introduce the concept of the oligo-vertex property for a finite set of matrices, and show that our algorithm runs in polynomial time if the given set of matrices has the oligo-vertex property. To the best of our knowledge, this is the first time such a property is introduced for a set of matrices. We derive several sufficient conditions for a set of matrices to have this property. On the other hand, we show that (P) is NP-hard for a pair of stochastic matrices or a pair of binary matrices, which implies that the oligo-vertex property is unlikely to hold for an arbitrary pair of \( n \times n \) integer matrices unless P=NP.

3. Numerical experiments demonstrate that our proposed algorithm is very efficient in practice, and has significant advantages over state-of-the-art global optimization software in solving large size instances.

The oligo-vertex property we propose may be of independent interest to readers. We want to point out some similarities between the oligo-vertex property and another important property for a set of matrices that is also concerned with long matrix products—the finiteness property. A finite set \( \Sigma \) of matrices is said to have the finiteness property if the joint spectral radius \( \rho(\Sigma) \) is equal to \( (\rho_k(T_{k-1}T_{k-2}\ldots T_0))^{1/k} \) with \( T_{k-1}, T_{k-2}, \ldots, T_0 \in \Sigma \) for some finite integer \( k \), where \( \rho(T) \) denotes the spectral radius of the matrix \( T \). The finiteness property has been studied extensively for different families of matrices [22, 18], as it has many implications on stability and stabilization of switched systems. The finiteness property and the oligo-vertex property both hold for the following sets of matrices: commuting matrices, any finite set of matrices with at most one matrix’s rank being greater than one [25], and a pair of \( 2 \times 2 \) binary matrices [19]. We suspect that there might be a deeper connection between these two properties. Finally we pose several open questions on the oligo-vertex property at the end of this paper.

The rest of the paper is organized as follows. In Section 2 we review results related to the problem we study, with a main focus on computational complexity. In Section 3 we first prove that (P) is NP-hard for a pair of stochastic matrices or binary matrices, and then introduce an exact algorithm for (P) and analyze its time complexity for general \( n \) and \( n = 2 \). In Section 4, we introduce the oligo-vertex property and present several sufficient conditions for a set of matrices to have the oligo-vertex property. In Section 5, we prove that a pair of \( 2 \times 2 \) binary matrices has the oligo-vertex property. We present some computational results in Section 6 and conclude in Section 7 with some open problems.
2 Related Work

Our problem aims to find the optimal switching rule of a discrete-time switched linear system without continuous control input. There have been a rich body of theoretical and computational results on optimal control of switched linear systems, such as finding optimal switching instants given a fixed switching sequence [40], minimizing the number of switches with known initial and final states [10], finding suboptimal policies [3], study of the exponential growth rates of the trajectories under different switching rules [16], and characterizing the value function of switched linear systems with linear and quadratic objectives [41]. We refer interested readers to the books [36, 23] and recent surveys [37, 42] for more details on switched linear systems. Finding the optimal switching sequence for a switched linear system also belongs to a broader class of problems called mixed-integer optimal control [34, 35] or optimal control of hybrid systems [2], which can be reformulated as a mixed-integer nonlinear optimization problem and solved by general mixed-integer optimization solvers.

We now survey computational complexity results related to the problem we study. Blondel and Tsitsiklis showed that the matrix mortality problem is undecidable for a pair of $48 \times 48$ integer matrices and the matrix $k$-mortality problem is NP-complete for a pair of $n \times n$ binary matrices with $n$ being an input parameter [6]. The complexity of the matrix $k$-mortality problem is however unknown when the matrix dimension $n$ is fixed. For the antibiotics time machine problem, Mira et al. used exhaustive search to find the optimal sequence of antibiotics for a small sized problem [26]. Tran and Yang showed that the antibiotics time machine problem is NP-hard when the number of matrices and the matrix dimension are both input parameters [38]. The antibiotics time machine can be also seen as a special finite-horizon discrete-time Markov decision process in which no state is observable. It has been shown in [30] that the finite-horizon unobservable Markov decision process is NP-hard. Therefore, our results identify several polynomially solvable cases of finite-horizon unobservable Markov decision processes. Computing the joint spectral radius for a finite set of matrices either exactly or approximately has been shown to be NP-hard [39], and has been a topic of active research [4, 31, 14]. The finiteness conjecture [22], which states that the finiteness property holds any set of real matrices, had remained a major open problem in the control community until early 2000s when a group of researchers showed that there exists a pair of $2 \times 2$ matrices that doesn’t have the finiteness property [8, 5, 21]. The first constructive counterexample for the finiteness conjecture was proposed in [14]. The finiteness conjecture was shown to be true for a pair of $2 \times 2$ binary matrices [19] and a finite set of matrices with at most one matrix’s rank being greater than one [25].

3 Computational Complexity and the Algorithm

3.1 Notations

We first introduce some notations that will be used throughout this paper. Let $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_-$ denote the sets of natural numbers (including 0), integers, real numbers, non-negative real numbers, and non-positive real numbers, respectively. We use $x_i$ to denote the $i$-th component of a given vector $x$. Let $\|x\|_\infty$ and $\|T\|_\infty$ denote the infinity norm of vector $x$ and matrix $T$, respectively. Given two positive integers $i, j$, let $[i : j]$ denote the set of integers $\{i, i + 1, \ldots, j\}$ if $i \leq j$ and $\emptyset$ if $i > j$. Given two scalar functions $f$ and $g$ defined on some subset of real numbers, we write $f(x) = O(g(x))$ as $x \to \infty$, if there exist $\alpha$ and $x_0 \in \mathbb{R}$ such that $|f(x)| \leq \alpha |g(x)|$ for all $x \geq x_0$. Given a set $S$, let $|S|$ denote the cardinality of $S$, conv$(S)$ denote the convex hull of $S$, int$(S)$ denote the interior of $S$, and $\partial S$ denote the boundary of $S$, respectively. Let ext$(S)$ denote the set of extreme points of a convex set $S$. Given a set $S \subseteq \mathbb{R}^n$ and a matrix $T \in \mathbb{R}^{n \times n}$, let...
be NP-complete [11]. Over these variables, each with three literals, can all be satisfied. The 3-SAT problem is known to whether there exists a truth assignment of several variables such that a given set of clauses defined over these variables, each with three literals, can all be satisfied. The 3-SAT problem is known to be NP-complete [11].

3.2 Complexity

Theorem 1. (P) is NP-hard for a pair of left (right) stochastic matrices and a linear function $f$.

Proof. We prove the result based on a reduction from the 3-SAT problem. A 3-SAT problem asks whether there exists a truth assignment of several variables such that a given set of clauses defined over these variables, each with three literals, can all be satisfied. The 3-SAT problem is known to be NP-complete [11].

Given an instance of the 3-SAT problem with $n$ variables $y_1, \ldots, y_n$ and $m$ clauses $C_1, \ldots, C_m$, we construct an instance of (P) with $\Sigma = \{A, B\}$ as follows. Matrices $A$ and $B$ are $m(2n+1) \times m(2n+1)$ adjacency matrices of two directed graphs $G_A$ and $G_B$, respectively. The construction of $G_A$ and $G_B$ will be explained in detail below. We set the total number of periods $K = n$. Let $e_k \in \mathbb{R}^{m(2n+1)}$ be a vector with the $k$-th entry being 1 and all other entries being 0. We set $x(0) = \sum_{j=1}^{m} e_j(2n+1)+1$ and $f(x) = c^\top x$ with $c = -\sum_{j=1}^{m} e_j(2n+1)$. We claim that the 3-SAT instance is satisfiable if and only if the optimal objective value of the constructed instance of (P) is $-m$.

Graph $G_A$ is constructed as follows. It contains $m(2n + 1)$ nodes, divided equally into $m$ groups, each group corresponding to a clause. There is no arc between nodes in different groups. Let $u_{j,1}, u_{j,2}, \ldots, u_{j,2n+1}$ be the $2n + 1$ nodes corresponding to clause $j$. The arcs among these nodes are as follows. Node $u_{j,2n+1}$ has a self loop. There is an arc from $u_{j,l+1}$ to $u_{j,l}$ for $l = [1 : 2n]$ unless literal $y_l$ is included in clause $C_j$; in that case, there will be an arc from node $u_{j,n+l+1}$ to node $u_{j,l}$. Graph $G_B$ is constructed similarly with the same set of nodes. There is an arc from $u_{j,l+1}$ to $u_{j,l}$ for $l = [1 : 2n]$ unless literal $y_l^c$ is included in clause $C_j$; in that case, there will be an arc from node $u_{j,n+l+1}$ to node $u_{j,l}$. An example for the clause $C_j = y_1 \lor y_3^c \lor y_4$ with a total of 4 variables is shown in Figure 2.

For $j \in [1 : n]$, let $A^j$ ($B^j$) be the adjacency matrix of the component of $G_A$ ($G_B$) corresponding to the $j$-th clause. Since each node has in-degree 1, each column of $A^j$ ($B^j$) has exactly one entry being 1, so $A^j$ ($B^j$) is a left stochastic matrix. We can associate each truth assignment of $y_1, \ldots, y_n$ with a sequence of matrices $T_0^j, \ldots, T_{n-1}^j$ with $T_t^j \in \{A^j, B^j\}$ for $t \in [0 : n-1]$. In particular, if $y_t$ is true (false), then $T_t^j$ is $A$ ($B$). Consider the product

$$[0, \ldots, 0, -1]T_{n-1}^j T_{n-2}^j \cdots T_0^j \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$  

It can be verified that this product is $-1(0)$ if any only if the truth assignment of $y_1, \ldots, y_n$ makes clause $j$ satisfied (unsatisfied).

Order the nodes of $G_A$ or $G_B$ lexicographically, i.e.,

$$u_{1,1}, u_{1,2}, \ldots, u_{1,2n+1}, u_{2,1}, \ldots, u_{2,2n+1}, \ldots, u_{m,2n+1}.$$  

Let $A$ and $B$ be the adjacency matrix of $G_A$ and $G_B$, respectively. Then both $A$ and $B$ are block
diagonal matrices with $m$ blocks of $(2n + 1) \times (2n + 1)$ matrices. In particular,

$$A = \begin{bmatrix} A^1 & A^2 & \cdots & A^m \\ & & & \\ & & & \\ & & & \end{bmatrix}, B = \begin{bmatrix} B^1 & B^2 & \cdots & B^m \\ & & & \\ & & & \\ & & & \end{bmatrix}. \quad (4)$$

Both $A$ and $B$ are left stochastic matrices. When $x(0) = \sum_{j=1}^{m} e_{j(2n+1)+1}$, $c = -\sum_{j=1}^{m} e_{j(2n+1)}$, $T_t \in \{ A, B \}$ for $t \in [0 : n - 1]$,

$$c^\top T_{n-1} \cdots T_0 x(0) = \sum_{j=1}^{m} [0, \cdots, 0, -1] T_{n-1}^j T_{n-2}^j \cdots T_0^j \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. $$

Therefore, there exists a truth assignment such that the 3-SAT instance is satisfied if and only if the optimal objective value of the constructed instance of (P) is $-m$. This reduction is done in time polynomial in $m$ and $n$.

To prove that (P) is NP-hard for a pair of right stochastic matrices, we can construct an instance of (P) in a similar way to the case of left stochastic matrices and show that there exists a truth assignment such that the 3-SAT instance is satisfied if and only if the optimal objective value of the constructed instance is $-m$. In particular, we let $x(0) = -\sum_{j=1}^{m} e_{j(2n+1)}$ (the vector $c$ in the instance of (P) with left stochastic matrices above), $f(x) = c^\top x$ with $c = \sum_{j=1}^{m} e_{j(2n+1)+1}$ (the initial vector $x(0)$ in the instance of (P) with left stochastic matrices above), and the two matrices be the transpose of the two matrices $A$ and $B$ defined in (4).

Since the matrices constructed in the proof of Theorem 1 are also binary matrices, we have the following result.

\[ \square \]
Corollary 1. (P) is NP-hard for a pair of binary matrices and a linear function $f$.

3.3 The Algorithm

In this section, we present a simple forward dynamic programming algorithm to solve (P) exactly, described in Algorithm 1. The critical step of Algorithm 1 is Step 6 which constructs $E_k$, the set of extreme points of $P_k(\Sigma, a)$, sequentially for $k = 0, 1, \ldots, K$.

Algorithm 1 A forward dynamic programming algorithm to solve (P).

1: **Input:** Matrices $\Sigma = \{A_1, \ldots, A_m\} \in \mathbb{Z}^{n \times n}$, initial vector $a \in \mathbb{Z}^n$, value oracle $f$, and positive integer $K$.
2: **Output:** A sequence of matrices $T_0, \ldots, T_{K-1} \in \Sigma$ that maximize $f(T_{K-1}T_{K-2} \cdots T_0a)$.
3: **Initialize:** Set $E_0 = \{a\}$.
4: for $k = 0, 1, \ldots, K-1$ do
5:   Set $F_i^k = A_i E_k$ for $i = 1, \ldots, m$.
6:   For each point $x \in \bigcup_{i=1}^m F_i^k$, check if $x$ is an extreme point of $\text{conv}(\bigcup_{i=1}^m F_i^k)$, by solving a linear program. Let $E_{k+1}$ be the set of all extreme points of $\text{conv}(\bigcup_{i=1}^m F_i^k)$.
7: end for
8: Find an $x^*(K) \in \text{arg max}\{f(x) \mid x \in E_K\}$ by enumeration.
9: Retrieve the optimal matrix sequence $T_{K-1}, T_{K-2}, \ldots, T_0$ from $x^*(K)$.

We specify the details of Step 6 later. In fact, Step 6 can be any algorithm that takes a set of points $S$ as input and output $\text{ext(}\text{conv}(S))$. There are several efficient algorithms to construct the convex hull of a set of points on the plane, more efficient than linear programs. It is, however, difficult to construct $\text{conv}(S)$ efficiently in higher dimensional space. The correctness of Algorithm 1 follows directly from the proposition below.

Proposition 1. Algorithm 1 solves (P) correctly.

Proof. First it is not difficult to show by induction that the set $E_k$ constructed in Algorithm 1 is the set of extreme points of $P_k(\Sigma, a)$ for each $k \in \{0 : K\}$. Since maximizing a convex function $f$ over a finite set $S$ is equivalent to maximizing $f$ over $\text{conv}(S)$ as well as maximizing $f$ over $\text{ext(}\text{conv}(S))$, (P) is equivalent to $\max\{f(x) \mid x \in P_K(\Sigma, a)\} = \max\{f(x) \mid x \in E_K\}$. Then the result follows.

Remark 1. The fact that we are maximizing a convex function in the objective is critical for the correctness of Algorithm 1. If we minimize $f(x(K))$ in (P) instead, then Algorithm 1 will not give the correct optimal solution in general.

We now specify the linear program in Step 6 of Algorithm 1. Given a finite set $S = \{p^1, \ldots, p^l\} \subseteq \mathbb{R}^n$, checking if a point $p^i \in S$ is an extreme point of $\text{conv}(S)$ can be done by solving the linear programming below.

$$
\begin{align}
    v^* &= \max (p^j)^\top z - z_0 \quad \text{(5a)}
    \\
    \text{s.t.} \quad (p^j)^\top z - z_0 &\leq 0, \quad i = 1, \ldots, l, i \neq j \quad \text{(5b)}
    \\
    (p^j)^\top z - z_0 &\leq 1. \quad \text{(5c)}
\end{align}
$$

Problem (5) is always feasible and bounded. If $v^* > 0$, then $p^j$ is an extreme point of $\text{ext}(S)$ and otherwise not. Problem (5) can be solved by various interior point methods in polynomial time,
for example Karmarkar’s algorithm. Recall that $M$ is the maximum absolute value of the entries of $A_1, \ldots, A_m$, and $a$.

**Proposition 2.** If Karmarkar’s algorithm is employed to solve the linear programs at Step 6, the running time of Algorithm 1 is $O(m^2 n^{4.5} (\log n + \log M) \sum_{k=0}^{K-1} kN_k(\Sigma)^2)$.

**Proof.** We first show that the sizes of all data in Algorithm 1 are polynomial in the problem input size, which is polynomial in $K, n,$ and $\log M$. To see this, for any integer $k \geq 0$,

$$\|x(k)\|_\infty = \max\{\|A_i x(k-1)\|_\infty \mid A_i \in \Sigma\} \leq \max\{\|A_i\|_\infty \mid A_i \in \Sigma\} \cdot \|x(k-1)\|_\infty$$

$$\leq (\max\{\|A_i\|_\infty \mid A_i \in \Sigma\})^k \cdot \|a\|_\infty \leq (nM)^k.$$

Therefore, the size of $x(k)$ is $O(n \log \|x(k)\|_\infty) = O(kn(\log n + \log M))$.

At Step 6 of iteration $k$, the number of operations of solving one linear program (5) with $S = \cup_{i=1}^m F_k$ using Karmarkar’s algorithm is $O(n^{3.5} L)$, where the input length $L = O(\sum_{i=1}^m |F_k| \log \|x(k)\|_\infty) = O(kmn(\log n + \log M)|E_k|)$. Since we need to solve $m|E_k|$ linear programs, one for each point in $S$, the running time of Step 6 is $m|E_k| O(n^{3.5} L) = O(km^2 n^{4.5} (\log n + \log M)|E_k|^2)$. At iteration $k$, Step 6 takes $O(mn^2)$ time, Step 8 takes $|E_k|$ queries to the value oracle of function $f$, and Step 9 can be performed in $K$ steps if a $m$-ary tree is used to store the values of $x(k)$ for each $k$. Therefore, the step with the dominating complexity is Step 6 and the overall running time of Algorithm 1 is $O(m^2 n^{4.5} (\log n + \log M) \sum_{k=0}^{K-1} k|E_k|^2)$. Since $|E_k| \leq N_k(\Sigma)$, the result follows.

### 3.3.1 Speeding up Algorithm 1 when $n = 2$

When $n = 2$, there are many efficient algorithms to construct the convex hull of a set of points directly, such as Graham’s scan and Jarvis’s march [9]. Graham’s scan constructs the convex hull of $l$ points on the plane in $O(l \log l)$ time [12]. With a similar analysis as in Proposition 2, we have the result below.

**Proposition 3.** When $n = 2$ and Graham’s scan is employed at Step 6 of Algorithm 1 to construct $E_{k+1}$, the running time of Algorithm 1 is $O(m \log m \sum_{k=0}^{K-1} N_k(\Sigma) + m \sum_{k=0}^{K-1} N_k(\Sigma) \log N_k(\Sigma))$.

### 4 Polynomially Solvable Cases

In this section, we focus on discovering conditions on a set of matrices for which (P) is polynomially solvable. Propositions 2 and 3 indicate that (P) is polynomially solvable if $N_k(\Sigma)$ is polynomial in $k$. This is the motivation that we introduce the concept of the oligo-vertex property in Section 1. Recall that a set of matrices $\Sigma$ has the oligo-vertex property if $N_k(\Sigma) = O(k^d)$ for some constant $d$. The following proposition gives the detailed time complexity of our algorithms for matrices with the oligo-vertex property, following directly from Proposition 2 and 3.

**Proposition 4.** If the set of matrices $\Sigma$ in (P) has the oligo-vertex property and $N_k(\Sigma) = O(k^d)$ for some constant $d$, then (P) can be solved in $O(m^2 n^{4.5} K^{2d+2} (\log n + \log M))$ time for general $n$ and in $O(m K^{2d+2} (\log m + \log K))$ time when $n = 2$.

Thus our focus in this section is to discover conditions for a set of matrices to have the oligo-vertex property. We introduce additional notations that will be used in the rest of the paper. Given a set of matrices $\Sigma = \{A_1, A_2, \ldots, A_m\} \subseteq \mathbb{R}^n$ and a vector $a \in \mathbb{R}^n$, define

$$X_k(\Sigma, a) = \{x(k) \mid x(k) = T_{k-1} \cdots T_0 a, T_j \in \Sigma, j \in [0 : k-1]\}$$

$$E_k(\Sigma, a) = \text{ext}(P_k(\Sigma, a))$$
for each integer $k \geq 0$. Recall that $P_k(\Sigma, a) = \text{conv}(X_k(\Sigma, a))$, $N_k(\Sigma, a) = |E_k(\Sigma, a)|$, and $N_k(\Sigma) = \sup_{a \in \mathbb{R}^n} \{N_k(\Sigma, a)\}$. Since $P_k(\Sigma, a)$ is the convex hull of at most $m^k$ points, both $N_k(\Sigma, a)$ and $N_k(\Sigma)$ are well defined and bounded above by $m^k$.

Some obvious cases that have the oligo-vertex property include a set $\Sigma$ of $m$ pairwise commuting matrices with constant $m$ (for which $N_k(\Sigma) = O(k^{m-1})$ since there are at most $(k+1)^{m-1}$ elements in $X_k(\Sigma, a)$), and a pair of projection matrices since there are at most $2k$ elements in $X_k(\Sigma, a)$.

**Proposition 5.** A set $\Sigma$ of $m$ matrices with at most one matrix with rank greater than one has the oligo-vertex property and $N_k(\Sigma) = O(mk)$.

**Proof.** Let $\Sigma = \{A_1, \ldots, A_m\}$. With loss of generality, assume that no $A_i$ is the zero matrix, and $A_1, A_2, \ldots, A_{m-1}$ are of rank one. Then for any $a \in \mathbb{R}^2$ the set $A_i P_k(\Sigma, a)$ contains at most two extreme points for $i = 1, \ldots, m-1$. For each integer $k \geq 0$, $P_{k+1}(\Sigma, a) = \text{conv}(\bigcup_{i=1}^m A_i P_k(\Sigma, a))$, so $N_{k+1}(\Sigma, a) \leq \sum_{i=1}^m |\text{ext}(A_i P_k(\Sigma, a))| \leq 2(m-1) + N_k(\Sigma, a)$. Then $N_{k+1}(\Sigma, a) \leq N_0(\Sigma, a) + 2k(m-1)$, so $N_k(\Sigma) = O(mk)$. □

**Proposition 6.** A set $\Sigma$ of two $2 \times 2$ matrices that share at least one common eigenvector has the oligo-vertex property and $N_k(\Sigma) = O(k)$.

**Proof.** If matrices $A$ and $B$ in $\Sigma$ share two eigenvectors, then they commute and there are at most $k+1$ different points in $X_k(\Sigma, a)$ for any $a$. Now suppose that $A$ and $B$ in $\Sigma$ share exactly one eigenvector $q_1$. Then $q_1$ must be a real vector. Assume the corresponding eigenvalues of $q_1$ in $A$ and $B$ are $\lambda_{11}$ and $\mu_{11}$, respectively. Since $q_1$ is a real vector, $\lambda_{11}$ and $\mu_{11}$ are both real-valued. Without loss of generality, assume $\|q_1\|_2 = 1$. Let $q_2 \in \mathbb{R}^2$ be a unit vector orthogonal to $q_1$. Consider the vector $A q_2$. Since $q_1$ and $q_2$ form a basis of $\mathbb{R}^2$, we have $A q_2 = \lambda_{12} q_1 + \lambda_{22} q_2$ for some $\lambda_{12}, \lambda_{22} \in \mathbb{R}$. Similarly, we have $B q_2 = \mu_{12} q_1 + \mu_{22} q_2$ for some $\mu_{12}, \mu_{22} \in \mathbb{R}$. Let $Q = [q_1 \quad q_2]$, $\Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ 0 & \lambda_{22} \end{bmatrix}$, $M = \begin{bmatrix} \mu_{11} & \mu_{12} \\ 0 & \mu_{22} \end{bmatrix}$. We have $\Lambda$ and $M$ as real matrices, $QQ^\top = I$, $A = Q \Lambda Q^\top$, and $B = Q M Q^\top$.

Any product of $k$ matrices with $A$ and $B$ can be written in the form of $A^{l_1} B^{m_1} A^{l_2} B^{m_2} \ldots A^{l_s} B^{m_s}$ with $l_1, m_1 \in \mathbb{N}$, $l_2, \ldots, l_s, m_1, \ldots, m_{s-1} > 0$ for some $s \geq 1$, and $\sum_{j=1}^s (l_j + m_j) = k$. We simplify the product as follows.

\[
A^{l_1} B^{m_1} A^{l_2} B^{m_2} \ldots A^{l_s} B^{m_s} = Q \begin{bmatrix} \lambda_{11}^{l_1} \cdots \lambda_{11}^{l_s} & \mu_{11}^{l_1} \cdots \mu_{11}^{l_s} \\ \lambda_{22}^{l_1} \cdots \lambda_{22}^{l_s} & \mu_{22}^{l_1} \cdots \mu_{22}^{l_s} \end{bmatrix} Q^\top,
\]

where $p = l_1 + \ldots + l_s$ and $*$ represents some real number. Let $\Pi_p$ be the set of all matrices in the form of $\begin{bmatrix} \lambda_{11}^p & \mu_{11}^p \\ 0 & \mu_{22}^p \end{bmatrix}$ calculated from a product of $k$ matrices with $p$ matrix $A$'s and $(k-p)$ matrix $B$'s. The set $\Pi_0$ contains one matrix in the form of $\begin{bmatrix} \mu_{11}^k \\ 0 \end{bmatrix}$. Call this matrix $C_0$. The set $\Pi_k$ contains one matrix in the form of $\begin{bmatrix} \lambda_{11}^k \\ 0 \end{bmatrix}$. Call this matrix $C_k$. For $1 \leq p \leq k-1$, any matrix in $\Pi_p$ can be represented as a convex combination of two matrices in $\Pi_p$, the ones with the smallest and largest $*$ entries. Call these two matrices $C_p$ and $D_p$. Then for any
p \in [1 : k - 1]$, the vector $x(k) = A^1 B^{m_1} A^2 B^{m_2} \cdots A^s B^{m_s} a$ with $\sum_{j=1}^{s} l_j = p$ can be represented by a convex combination of $C_p a$ and $D_p a$. Hence $P_k(\Sigma, a) = \text{conv}\{C_0 a, C_1 a, D_1 a, C_2 a, D_2 a, \ldots, C_k a\}$. Therefore $N_k(\Sigma, a) \leq 2k$ and $N_k(\Sigma) = O(k)$.

**Remark 2.** Each right stochastic matrix has an eigenvector $(1, 1)^\top$. Therefore, any pair of $2 \times 2$ right stochastic matrices has the oligo-vertex property and the corresponding problem (P) is polynomially solvable.

Finally we present a lemma showing that the oligo-vertex property is invariant under any similarity transformation.

**Lemma 1.** A finite set of $n \times n$ matrices $\Sigma$ has the oligo-vertex property if and only if $SS^{-1}$ has the oligo-vertex property for any nonsingular real matrix $S$.

**Proof.** It suffices to show that $N_k(\Sigma) = N_k(SS^{-1})$. We claim that $P_k(\Sigma, a) = P_k(SS^{-1}, Sa)$ for any $a \in \mathbb{R}^n$. To see this, note that any extreme point $p$ of $P_k(\Sigma, a)$ can be written as $p = T_{k-1}T_{k-2}\cdots T_0 a$ with $T_j \in \Sigma$ or $j \in [0 : k - 1]$. Then

$$p = T_{k-1}T_{k-2}\cdots T_0 a = S^{-1}(ST_{k-1}S^{-1})(ST_{k-2}S^{-1})\cdots(ST_0 S^{-1})Sa.$$

We have $p \in S^{-1}P_k(SS^{-1}, Sa)$. Therefore, $P_k(\Sigma, a) \subseteq S^{-1}P_k(SS^{-1}, Sa)$. Similarly, we can show that $P_k(\Sigma, a) \supseteq S^{-1}P_k(SS^{-1}, Sa)$, so $P_k(\Sigma, a) = S^{-1}P_k(SS^{-1}, Sa)$. Since $S$ is nonsingular, the number of extreme points of $P_k(\Sigma, a)$ equals the number of extreme points of $P_k(SS^{-1}, Sa)$, i.e., $N_k(\Sigma, a) = N_k(SS^{-1}, Sa)$. Thus $N_k(\Sigma) = \sup_{a \in \mathbb{R}^n} N_k(\Sigma, a) = \sup_{a \in \mathbb{R}^n} N_k(SS^{-1}, Sa) \leq N_k(SS^{-1})$. By symmetry, we can show that $N_k(SS^{-1}) \leq N_k(\Sigma)$. Therefore, $N_k(\Sigma) = N_k(SS^{-1})$.

**5 The $2 \times 2$ Binary Matrices**

Our main result in this section is the following theorem.

**Theorem 2.** A pair of $2 \times 2$ binary matrices has the oligo-vertex property.

The seemingly innocent looking statement above is the most difficult to prove in this paper. In fact, we are unable to provide a unified argument for all $2 \times 2$ binary matrices. This is not too surprising, however, since to the best of our knowledge there is no unified argument to show that any pair of $2 \times 2$ binary matrices has the finiteness property either [19]. We hope that the techniques we develop in this paper can be useful in proving the oligo-vertex property for other matrices in the future.

There are a total of 16 binary matrices, resulting in a total of 120 different pairs of $2 \times 2$ binary matrices. To prove Theorem 2, we first show that the result holds for most of the 120 pairs, and then provide separate proofs for each of the remaining pairs. Among the 16 binary matrices, one matrix has rank zero, nine matrices have rank one, and six matrices have rank two. The pair of matrices has the oligo-vertex property if one matrix is the zero or identity matrix. According to Proposition 5, the pair of matrices has the oligo-vertex property if one matrix is singular. Therefore, only the following five binary matrices of rank two give rise to interesting pairs:

$$A_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad A_5 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$
The five matrices above give rise to ten different pairs of binary matrices. Observe that
\[ A_1A_1A_1^{-1} = A_1, \ A_1A_2A_1^{-1} = A_3, \ A_1A_4A_1^{-1} = A_5, \ A_2A_5A_2^{-1} = A_4. \]

Then by Lemma 1, we can group the ten pairs of matrices into the following five clusters:

1. \( \{A_1, A_2\}, \{A_1, A_3\} \)
2. \( \{A_1, A_4\}, \{A_1, A_5\} \)
3. \( \{A_2, A_3\} \)
4. \( \{A_4, A_5\} \)
5. \( \{A_2, A_4\}, \{A_3, A_5\}, \{A_2, A_5\}, \{A_3, A_4\} \),

and it suffices to show that one pair of matrices within each cluster has the oligo-vertex property. In the rest of this section, we are going to show separately that each of the following five pairs of matrices has the oligo-vertex property.

\[ \Sigma_1 = \{A_1, A_2\}, \Sigma_2 = \{A_1, A_4\}, \Sigma_3 = \{A_2, A_3\}, \Sigma_4 = \{A_4, A_5\}, \Sigma_5 = \{A_2, A_4\}. \]

We first present in the table below a complete description of how \( N_k(\Sigma, a) \) grows with \( k \) for the five pairs of matrices, according to the location of the initial vector \( a \).

<table>
<thead>
<tr>
<th>( a \in \mathcal{Q}_1 \cup \mathcal{Q}_3 )</th>
<th>( \Sigma_1 )</th>
<th>( \Sigma_2 )</th>
<th>( \Sigma_3 )</th>
<th>( \Sigma_4 )</th>
<th>( \Sigma_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( o(k^2) )</td>
<td>( O(k) )</td>
<td>( O(k) )</td>
<td>( O(k) )</td>
<td>( O(k) )</td>
<td></td>
</tr>
<tr>
<td>( a \in \text{int}(\mathcal{Q}_2) \cup \text{int}(\mathcal{Q}_4) )</td>
<td>( O(k^4) )</td>
<td>( O(k) )</td>
<td>( O(k^2) )</td>
<td>( O(k^2) )</td>
<td>( O(k^2) )</td>
</tr>
</tbody>
</table>

Table 1: The number of extreme points \( N_k(\Sigma, a) \)

The results in Table 1 show that the number of extreme points of \( P_k(\Sigma, a) \) grows linearly with \( k \) when the initial vector is in the first or the third quadrant for most pairs of binary matrices except \( \Sigma_1 \).

Example 2. Figure 3 illustrates how the number of extreme points \( N_k(\Sigma_1, a) \) changes with \( k \) given different initial vector \( a \)’s. For the chosen \( a \)’s, the growth is at most linear in \( k \) for \( k \leq 40 \).

To prove the results in Table 1 we first introduce a few notations that will be used in the rest of this section. Given a pair \( \Sigma \) of matrices and a vector \( a \in \mathbb{R}^2 \), we divide the set of extreme points \( E_k(\Sigma, a) \) of \( P_k(\Sigma, a) \) into five groups.

Definition 2. Let \( E^i_k(\Sigma, a) \) be the set of extreme points of \( P_k(\Sigma, a) \) that are maximizers of the linear program \( \max \{cx \mid x \in P_k(\Sigma, a)\} \) for some \( c \in \text{int}(\mathcal{Q}_i) \), for \( i = 1, 2, 3, 4 \). Let \( E^0_k(\Sigma, a) \) be the set of extreme points of \( P_k(\Sigma, a) \) that are maximizers of the linear programs \( \max \{cx \mid x \in P_k(\Sigma, a)\} \) where \( c \in \{(1, 0), (0, 1), (-1, 0), (0, -1)\} \).

Then
\[ E_k(\Sigma, a) = \bigcup_{i=0}^4 E^i_k(\Sigma, a) \text{ and } N_k(\Sigma, a) \leq \sum_{i=0}^4 |E^i_k(\Sigma, a)|. \quad (8) \]

Example 3. Figure 4 illustrates the polytopes \( P_k(\Sigma_3, a) \) and the sets of extreme points \( E^i_k(\Sigma_3, a) \) for \( i \in [0 : 4] \) with \( a = (2, 1)^\top \), for \( k = 5 \) and \( k = 7 \).
Figure 3: The number of extreme points $N_k(\Sigma_1, a)$ given different initial vector $a$'s.

Figure 4: Examples of polytopes $P_k(\Sigma_3, a)$ and associated sets of extreme points $E_i^k(\Sigma_3, a)$ for $i \in [0:4]$.

5.1 $\Sigma_1 = \{A_1, A_2\}$

Proposition 7. The pair $\Sigma_1$ has the oligo-vertex property and $N_k(\Sigma_1) = O(k^4)$.

Proposition 7 is an immediate consequence of the following propositions.

Proposition 8. For any $a \in \text{int}(Q_1) \cup \text{int}(Q_3)$, $N_k(\Sigma_1, a) = O(k^2)$.

Proposition 9. For any $a \in \partial Q_1 \cup \partial Q_3$, $N_k(\Sigma_1, a) = O(k^2)$.

Proposition 10. For any $a \in \text{int}(Q_2) \cup \text{int}(Q_4)$, $N_k(\Sigma_1, a) = O(k^4)$.

We first focus on proving Proposition 8. Our strategy is to bound the cardinality of $E_i^k(\Sigma_1, a)$ for each $i$. Then according to (8), $N_k(\Sigma_1)$ will be bounded as well.

Lemma 2. For any $a \in \text{int}(Q_1)$ and integer $k \geq 2$, $|E_i^1(\Sigma_1, a)| \leq k + 2$.

Proof. To simplify the notations, we write $E_i^k$ and $P_k$ instead of $E_i^k(\Sigma_1, a)$ and $P_k(\Sigma_1, a)$ respectively in the rest of the proof. We claim that $|E_k^l| \leq |E_{k-1}^l| + 1$ for $k \geq 2$. Then $|E_k^1| \leq |E_1^1| + (k - 1)$.
2 + (k - 1) = k + 1. To prove the claim, we first show that $E_k^1 \subseteq A_1E_{k-1}^1 \cup A_2E_{k-1}^1$. Note that

$$\max \{ cx \mid x \in P_k\} = \max \{ \max \{ cA_1 x \mid x \in P_{k-1}\}, \max \{ cA_2 x \mid x \in P_{k-1}\} \}. \quad (9)$$

Given $c \in \text{int}(Q_1)$, both $cA_1$ and $cA_2$ are in the interior of $Q_1$, so the maximizers of linear programs on the right are in the set $E_{k-1}^1$. Therefore, $E_k^1 \subseteq A_1E_{k-1}^1 \cup A_2E_{k-1}^1$.

Next we show that some points in $A_1E_{k-1}^1 \cup A_2E_{k-1}^1$ cannot belong to $E_k^1$. Let $p = (p_1, p_2)^\top \in E_{k-1}^1$ be the maximizer of the linear program $\max \{ x_1 + x_2 \mid x \in P_{k-1}\}$ with the smallest $x_2$-coordinate. Note that there is no other point in $E_{k-1}^1$ whose $x_2$-coordinate is $p_2$. Otherwise suppose that there is such a point $p'$. The fact that $p$ is the maximizer of $\max \{ x_1 + x_2 \mid x \in P_{k-1}\}$ implies $p_1 > p'_1$. Then $cp' < cp$ for any $c \in \text{int}(Q_1)$, which contradicts that $p' \in E_{k-1}^1$. Now we can partition $E_{k-1}^1$ into three sets $S_1 = \{ x \mid x \in E_{k-1}^1, x_2 > p_2 \}$, $S_2 = \{ p \}$, and $S_3 = \{ x \mid x \in E_{k-1}^1, x_2 < p_2 \}$. Then $E_k^1 \subseteq A_1S_1 \cup A_1S_2 \cup A_1S_3 \cup A_2S_1 \cup A_2S_2 \cup A_2S_3$. We show below that the points in $A_1S_1$ or $A_2S_3$ cannot be in $E_k^1$.

First consider any point $x \in S_1$.

- If $x_1 < x_2$, we have $cA_2x - cA_1x = c_1x_1 + c_2(x_2 - x_1) > 0$.
- Suppose $x_1 \geq x_2$ and $c_1 < c_2$. Since $x_1 + x_2 \leq p_1 + p_2$, we have $x_1 - p_1 \leq p_2 - x_2 < 0$. Therefore, $cA_1p - cA_1x = c_1(p_2 - x_2) + c_2(p_1 - x_1) \geq c_1(x_1 - p_1) + c_2(p_1 - x_1) = (c_1 - c_2)(x_1 - p_1) > 0$.
- Suppose $x_1 \geq x_2$ and $c_1 \geq c_2$. Since $x_1 + x_2 \leq p_1 + p_2$, $p_2 - x_1 \geq x_2 - p_1$. Since $x_1 \geq x_2 > p_2$ and $x_1 + x_2 \leq p_1 + p_2$, we have $p_1 \geq x_2$. Then $cA_2p - cA_1x = c_1p_1 + c_1(p_2 - x_2) + c_2(p_2 - x_1) \geq c_1p_1 + c_1(p_2 - x_2) + c_2(p_2 - x_1) = (c_1 - c_2)(p_1 - x_2) + c_1p_2 > 0$.

Therefore, $A_1x \in A_1S_1$ cannot be a maximizer of linear program \([9]\) with $c \in \text{int}(Q_1)$.

Now consider any point $x \in S_3$. Since $p_2 - x_2 > 0$ and $p_1 + p_2 \geq x_1 + x_2$, $cA_2p - cA_2x = c_1(p_1 + p_2 - x_1 - x_2) + c_2(p_2 - x_2) > 0$. Therefore, $A_2x \in A_2S_3$ cannot be a maximizer of linear program \([9]\) with $c \in \text{int}(Q_1)$. Hence, $|E_k^1| \leq |A_1S_1| + |A_1S_2| + |A_2S_1| + |A_2S_2| = |S_1| + |S_3| + |S_2| = |E_{k-1}^1| + 1$. \(\square\)

**Lemma 3.** For any $a \in \text{int}(Q_1)$ and integer $k \geq 2$, $|E_k^3(a, 1)| \leq 2$.

**Proof.** To simplify the notations, we write $E_k^3$ instead of $E_k^3(a, 1)$ in the rest of the proof. Let $a = (a_1, a_2)^\top \in \text{int}(Q_1)$. Assume that $a_1 \leq a_2$. The case in which $a_1 > a_2$ can be proved similarly.

We show below by induction that $E_k^3 \subseteq \{ A_t^1a, A_t^{-1}A_2A_1a \}$ for any $k \geq 2$. For the base case $k = 2$, given any $c \in \text{int}(Q_3)$, $cA_1^2a - cA_2^2a = -2c_2a_2 > 0, cA_1^2a - cA_1A_2a = c_1(a_1 - a_2) - c_2a_1 > 0$. Hence, $E_2^3 \subseteq \{ A_t^1a, A_2A_1a \}$.

Now suppose that $E_t^3 \subseteq \{ A_t^1a, A_t^{-1}A_2A_1a \}$ for some $t \geq 2$. We want to show that $E_{t+1}^3 \subseteq \{ A_{t+1}^1a, A_{t-1}^{-1}A_2A_1a \}$. We assume that $t$ is even (a similar argument can be used to prove the result when $t$ is odd). Similar to the proof of \([9]\) in Lemma 2, we have $E_k^3 \subseteq A_1E_{k-1}^1 \cup A_2E_{k-1}^1$ for $k \geq 2$. Then by the induction hypothesis, we have $E_{t+1}^3 \subseteq \{ A_{t+1}^1a, A_{t-1}^{-1}A_2A_1a, A_2A_{t}^1a, A_2A_{t}^{-1}A_2A_1a \}$ since $t$ is even, $A_{t}^1a = a$ and $A_{t-1}^{-2}A_2A_1a = (a_1 + a_2, a_1)^\top$. For any $c \in \text{int}(Q_3)$, $cA_{t+1}^1a - cA_{t-1}^{-1}A_2A_1a = -c_1a_1 + c_2(a_1 - a_2) > 0$, and $cA_{t+1}^1a - cA_{t-2}^{-2}A_2A_1a = -2c_1a_1 > 0$.

Hence, $E_{t+1}^3 \subseteq \{ A_{t+1}^1a, A_{t-1}^{-1}A_2A_1a \}$. We conclude that $E_k^3 \leq 2$ for any integer $k \geq 2$. \(\square\)

**Lemma 4.** For any $a \in \text{int}(Q_1)$ and integer $k \geq 2$, $|E_k^4(a, 1, 1)| \leq E_k^4(a, 1) + E_k^1(a, 1) + 2$ and $|E_k^3(a, 1)| \leq E_k^3(a, 1)$.

**Proof.** To simplify the notations, we omit the dependence of $\Sigma_1$ and $a$ in the rest of the proof. We first prove that $|E_k^4| \leq |E_{k-1}^4| + |E_{k-1}^1| + 2$. Note that

$$\max \{ cx \mid x \in P_k \} = \max \{ \max \{ cA_1A_1 x \mid x \in P_{k-2} \}, \max \{ cA_1A_2 x \mid x \in P_{k-2} \}, \max \{ cA_2A_1 x \mid x \in P_{k-2} \}, \max \{ cA_2A_2 x \mid x \in P_{k-2} \} \}.$$
Since $P_{k-2} \subseteq \text{int}(Q_1)$, for any $c$ with $c_1 > 0$ and $c_2 < 0$ and $x \in P_{k-2}$, $cA_2^2x = (c_1, 2c_1 + c_2)x > (c_1, c_2)x = cA_2^2x$, $cA_2^2x = (c_1, 2c_1 + c_2)x > (c_1, c_2 + c_2)x = cA_1A_2x$. Therefore, $\max \{cx \mid x \in P_k\} = \max \{\max \{cA_2A_1x \mid x \in P_{k-2}\}, \max \{cA_2A_2x \mid x \in P_{k-2}\}\} = \max \{cA_2x \mid x \in P_{k-1}\}$. Now that $cA_2 = (c_1, c_1 + c_2)$ is a vector in the first or the fourth quadrant, the maximizers of $\max \{cx \mid x \in P_k\}$ must be in $A_2E_{k-1}^{1} \cup A_2E_{k-1}^{2} \cup A_2S$, where $S$ is the set of extreme points of $P_{k-1}$ that are maximizers of $\max \{x_1 \mid x \in P_{k-1}\}$. Therefore, $|E_k^2| \leq |A_2E_{k-1}^{1}| + |A_2E_{k-1}^{1}| + |A_2S| \leq |E_{k-1}^{1}| + |E_{k-1}^{1}| + 2$.

To prove that $|E_{k-1}^{1}| \leq |E_{k-1}^{1}|$, consider $c = (c_1, c_2)$ with $c_1 < 0$ and $c_2 > 0$. For any $x \in P_{k-2}$,

$$cA_1A_2x = (c_2, c_1 + c_2)x > (c_1, c_2, c_1)x = c^TA_2A_1x,$$
$$cA_1A_2x = (c_2, c_1 + c_2)x > (c_1, 2c_1 + c_2)x = c^TA_2A_2x.$$

Thus we have $\max \{cx \mid x \in P_k\} = \max \{\max \{cA_1A_1x \mid x \in P_{k-2}\}, \max \{cA_1A_2x \mid x \in P_{k-2}\}\} = \max \{cA_1x \mid x \in P_{k-1}\}$. Since $cA_1 = (c_2, c_1)$ is a vector in the interior of the fourth quadrant, the optimal solutions of $\max \{cx \mid x \in P_k\}$ must be in $A_1E_{k-1}^{1}$. Therefore, $|E_{k-1}^{2}| \leq |E_{k-1}^{1}|$. □

Now we are ready to prove Proposition 8.

Proof of Proposition 8 We only need to prove the case where $a \in \text{int}(Q_1)$. When $a \in \text{int}(Q_3)$, it is easy to verify that $N_k(\Sigma_1, a) = N_k(\Sigma_2, -a)$. By Lemma 2 and Lemma 3, we have $|E_k^3(\Sigma_1, a)| \leq k + 2$ and $|E_k^3(\Sigma_1, a)| \leq 2$ for any $a \in \text{int}(Q_1)$ and integer $k \geq 2$. By Lemma 4, for any $a \in \text{int}(Q_1)$ and integer $k \geq 3$, $|E_k(\Sigma_1, a)| \leq |E_k^3(\Sigma_1, a)| + |E_k^4(\Sigma_1, a)| + 2$ \leq $|E_k^3(\Sigma_1, a)| + (k + 2) \leq |E_k^3(\Sigma_1, a)| + \frac{1}{2}k^2 + \frac{5}{2}k - 3$. Therefore, $N_k(\Sigma_1, a) \leq |E_k(\Sigma_1, a)| + |E_k^3(\Sigma_1, a)| + |E_k^4(\Sigma_1, a)| = O(k^2)$. □

The conclusion $N_k(\Sigma_1, a) = O(k^2)$ can be easily extended to the case where $a$ is on the boundary of the first or third quadrant.

Proof of Proposition 9 We only need to prove the case where $a \in \partial Q_1$. The case where $a \in \partial Q_3$ follows from the fact $N_k(\Sigma_1, a) = N_k(\Sigma_2, -a)$. We first prove the result when $a$ is on the positive $x_1$-axis. Without loss of generality, assume that $a = (1, 0)^T$. We claim that for any integer $k \geq 3$,

$$X_k(\Sigma_1, (1, 0)^T) = X_{k-2}(\Sigma_1, (1, 1)^T) \cup \{(1, 0)^T, (0, 1)^T\}.$$

To see this, consider any value of $x(k)$ in $X_k(\Sigma_1, (1, 0)^T)$ that is different from $(1, 0)^T$ and $(0, 1)^T$. Since $A_1^3a = (0, 1)^T$ for odd integer $t \geq 1$, $A_1^3a = (0, 1)^T$ for even integer $t \geq 1$, $A_2^3a = (0, 1)^T$ for any integer $t \geq 1$, and $A_2A_1a = (1, 1)^T$. For $x(k)$ to take a value different from $(0, 1)^T$ and $(1, 0)^T$, $x(k)$ must be in the form of $T_{k-1} \cdots T_1x(l)$ with $T_j \in \Sigma_1$ for $j \in \{l : k - 1\}$ and $x(l) = (1, 1)^T$ for some $l \geq 2$. But when $x(l) = (1, 1)^T$, we have $A_1^3x(l) = x(l)$ for any integer $l \geq 1$. Then $x(k) = T_{k-1} \cdots T_1A_1^3x(l)$, which is a point in $X_{k-2}(\Sigma_1, (1, 1)^T)$. Thus $X_k(\Sigma_1, (1, 0)^T) \subseteq X_{k-2}(\Sigma_1, (1, 1)^T) \cup \{(1, 0)^T, (0, 1)^T\}$. On the other hand, given a point in $X_{k-2}(\Sigma_1, (1, 1)^T)$ written in the form of $T_{k-3} \cdots T_0(1, 1)^T$ with $T_j \in \Sigma_1$ for $j \in \{l : k - 3\}$, we can also write it in the form of $T_{k-3} \cdots T_0A_2A_1(1, 0)^T$. Thus $X_k(\Sigma_1, (1, 0)^T) \supseteq X_{k-2}(\Sigma_1, (1, 1)^T) \cup \{(1, 0)^T, (0, 1)^T\}$. Therefore, $N_k(\Sigma_1, (1, 0)^T) = N_{k-2}(\Sigma_1, (1, 1)^T) + 2 = O(k^2)$. The last equality follows from Proposition 8. The case where $a$ is on the positive $x_2$-axis can be proved similarly. □

We proceed to prove Proposition 10. Let $X_k^4(\Sigma_1, a)$ be the set of points in $X_k(\Sigma_1, a)$ that are in the interior of the second or fourth quadrant, i.e.,

$$X_k^4(\Sigma_1, a) = X_k(\Sigma_1, a) \cap (\text{int}(Q_2) \cup \text{int}(Q_4)).$$
Lemma 5. For any $a \in \text{int}(Q_4)$ and integer $k \geq 2$, $X_k^{2,4}(\Sigma_1, a)$ contains no more than $4k + 4$ points.

Proof. Without loss of generality, assume $a = (1, a_2)^\top$ with $a_2 < 0$. Let $u_0 = \max\{1, -a_2\}$ and $v_0 = \min\{1, -a_2\}$. Define the following sequence of non-negative numbers recursively $u_j = \max\{v_{j-1}, u_{j-1} - v_{j-1}\}$ and $v_j = \min\{v_{j-1}, u_{j-1} - v_{j-1}\}$ for $j \in [1 : k]$. For each $t \in [0 : k]$, define $S_t = \{(u_t, -v_t)^\top, (-u_t, v_t)^\top, (v_t, -u_t)^\top, (-v_t, u_t)^\top\}$. Given any $s^k \in X_k^{2,4}(\Sigma_1, a)$, assume that $s^k = T_{k-1} \cdots T_0 a$ with $T_j \in \Sigma_1$ for $j \in [0 : k-1]$.

We claim that for any integer $k \geq 0$, if $t$ out of the $k$ matrices $T_0, \cdots, T_{k-1}$ are $A_2$, then $s^k \in S_t$. We prove the claim by induction on $k$. First consider the base case $k = 0$. If $|a_2| \geq 1$, then $u_0 = -a_2$ and $v_0 = 1$, so $s^k = a = (v_0, -u_0)^\top \in S_0$. If $|a_2| < 1$, then $u_0 = 1$ and $v_0 = -a_2$, so $s^k = a = (u_0, -v_0)^\top \in S_0$. Now suppose that the claim holds for integer $k = l \geq 0$. Specifically, $s^l = T_{l-1} \cdots T_0 a \in S_l$ if $t \in [0 : l]$ out of the $l$ matrices $T_0, \cdots, T_{l-1}$ are $A_2$. We want to prove that any point $s^{l+1} = T_{l+1} \cdots T_0 a$ in $X_{l+1}^{2,4}(\Sigma_1, a)$ also belongs to $S_l$, if $t \in [0 : l+1]$ out of the $l+1$ matrices $T_0, \cdots, T_{l+1}$ are $A_2$. If $T_{l+1} = A_1$, then $t$ out of the $l$ matrices $T_1, \cdots, T_0$ are $A_2$. Based on the induction hypothesis, the point $s = T_1 \cdots T_0 a \in S_{l-1}$. The set $S_l$ contains four points. We consider one case $s = (u_{t-1}, -v_{t-1})^\top$ here, and the result for the other cases can be proved similarly. We have $s^{l+1} = A_2 s = (u_{t-1} - v_{t-1}, -v_{t-1})^\top$. Since $s^{l+1}$ is in the interior of second or fourth quadrant and $-v_{t-1} < 0$, we must have $u_{t-1} - v_{t-1} > 0$. If $v_{t-1} \geq u_{t-1} - v_{t-1}$, then $u_t = v_{t-1}, v_t = u_{t-1} - v_{t-1}$, and $s^{l+1} = (v_t, -u_t)^\top \in S_l$. If $v_{t-1} < u_{t-1} - v_{t-1}$, then $u_t = u_{t-1} - v_{t-1}, v_t = v_{t-1}$, and $s^{l+1} = (u_t, -v_t)^\top \in S_l$. With the claim, we conclude that $X_k^{2,4}(\Sigma_1, a)$ contains at most $4k + 4$ different points.

Proof of Proposition [10] We omit the dependence of $\Sigma_1$ in the rest of the proof to simplify the notation. Given a set $S \subseteq \mathbb{R}^2$, define $X_k(S) = \cup_{a \in S} X_k^{2,4}(a)$.

First note that for any $x$ in the first (third) quadrant, $A_1 x$ and $A_2 x$ are both in the first (third) quadrant. Thus the points in $X_0^{2,4}(a)$ can only be linear transformations of points in $X_1^{2,4}(a)$ under $A_1$ or $A_2$. In addition, for any $x$ in the second or fourth quadrant, $A_1 x$ is also in the second or fourth quadrant. Therefore, for any integer $i \geq 0$, $A_1 X_i^{2,4}(a) \cup A_2 X_i^{2,4}(a) = X_{i+1}^{2,4}(a) \cup (A_2 X_i^{2,4}(a) \cap (Q_1 \cup Q_3))$. Given any $a$ in the interior of the second quadrant, we have

$$X_k(a) = X_k(X_0^{2,4}(a)) = X_{k-1}(A_1 X_0^{2,4}(a) \cup A_2 X_0^{2,4}(a))$$
$$= X_{k-1}(X_1^{2,4}(a)) \cup X_{k-1}(A_2 X_0^{2,4}(a) \cap (Q_1 \cup Q_3))$$
$$= (X_{k-2}(X_2^{2,4}(a)) \cup X_{k-2}(A_2 X_1^{2,4}(a) \cap (Q_1 \cup Q_3))) \cup X_{k-1}(A_2 X_0^{2,4}(a) \cap (Q_1 \cup Q_3))$$
$$\cdots$$
$$= X_l(X_{k-l}^{2,4}(a)) \cup \bigcup_{j=l}^{k-1} X_j(A_2 X_{k-j}^{2,4}(a) \cap (Q_1 \cup Q_3))$$

for any $l \geq 1$.

On the other hand, for any $x$ in the first or third quadrant, we have shown that there exists some integer $k_0$ and $\alpha > 0$ such that $N_k(x) \leq \alpha k^2$ for any integer $k \geq k_0$. Setting $l = k_0$ in...
Proposition 11. We have $X_k(a) = X_{k_0} (X_{k-4}^{2,4} (a)) \cup \bigcup_{j=k_0}^{k-1} X_j (A_{2} X_{k-4}^{2,4} (a) \cap (Q_1 \cup Q_3))$. Therefore,

$$N_k(a) \leq |X_{k_0} (X_{k-4}^{2,4} (a))| + \sum_{j=k_0}^{k-1} \sum_{x \in A_{2} X_{k-4}^{2,4} (a) \cap (Q_1 \cup Q_3)} N_j(x)$$

$$\leq \sum_{x \in X_{k-4}^{2,4} (a)} |X_{k_0} (x)| + \sum_{j=k_0}^{k-1} |A_{2} X_{k-4}^{2,4} (a)| \alpha_j^2$$

$$\leq |X_{k-4}^{2,4} (a)| 2^{k_0} + \sum_{j=k_0}^{k-1} |X_{k-4}^{2,4} (a)| \cdot \alpha_j^2$$

$$\leq (4k - 4k_0 + 4) 2^{k_0} + \sum_{j=k_0}^{k-1} (4k - 4j) \alpha_j^2 \leq \beta k^4,$$

for some constant $\beta$. The second last inequality follows from Lemma 5. Therefore $N_k(a) = O(k^4)$.

5.2 $\Sigma_2 = \{A_1, A_4\}$

In this section, we will prove that $N_k(\Sigma_2) = O(k)$.

Lemma 6. Given any polytope $P \subseteq \mathbb{R} \times \mathbb{R}_+$ or $P \subseteq \mathbb{R} \times \mathbb{R}_-$, the number of extreme points of $\text{conv}(P \cup A_2 P)$ is at most two more than the number of extreme points of $P$.

Proof. We first prove the case where $P \subseteq \mathbb{R} \times \mathbb{R}_+$. The result is easy to show if $P$ is a singleton or a line segment. Now suppose $P$ is full dimensional. Let $r = (r_1, r_2)^\top$ be the extreme point of $P$ with the largest $x_2$-coordinate; if there are two such extreme points, let $r$ be the one with a larger $x_1$-coordinate. Similarly, let $s = (s_1, s_2)^\top$ be the extreme point of $P$ with the smallest $x_2$-coordinate; let $s$ be the one with a larger $x_1$-coordinate if there are two such extreme points. Divide the extreme points of $P$ other than $r$ and $s$ into two sets: (1) Set $Q_1$ consisting of extreme points visited if we walk clockwise along the boundary of $P$ from $s$ to $r$; (2) Set $Q_2$ consisting of extreme points visited if we walk clockwise along the boundary of $P$ from $r$ to $s$. Let $R = \{r, s, A_{2}r, A_{2}s\}$. Since $\text{ext}(P) = Q_1 \cup Q_2 \cup \{r, s\}$, the possible extreme points of $\text{conv}(P \cup A_2 P)$ are among $Q_1$, $Q_2$, $A_{2}Q_1$, $A_{2}Q_2$, and $R$.

We claim that any point in $Q_2$ can be represented as a convex combination of points in $Q_1 \cup A_{2}Q_2 \cup R$. To see this, first consider any point $p = (p_1, p_2)^\top \in Q_2$. By the definition of $Q_2$, we have $p_2 > 0$ and there exists a point $h = (h_1, h_2)^\top$ on the line segment connecting $r$ and $s$ such that $h_1 < p_1$ and $h_2 = p_2$. See the illustration in Figure 5. We can verify that $p = \lambda A_{2}p + (1 - \lambda) h$ with $\lambda = \frac{p_1 - h_1}{p_2 - h_2} \in (0, 1)$. Thus $p$ can be represented as a convex combination of $A_{2}p$ and $h$. Since $h$ can also be represented by a convex combination of $r$ and $s$, $p$ can be represented as a convex combination of $A_{2}p$, $r$, and $s$. Therefore, we show that any point in $Q_2$ is a convex combination of points in $Q_1 \cup A_{2}Q_2 \cup R$.

Similarly, we can show that any point in $A_{2}Q_1$ is a convex combination of points in $Q_1 \cup A_{2}Q_2 \cup R$. Then we have $\text{conv}(P \cup A_2 P) = \text{conv}(Q_1 \cup A_{2}Q_2 \cup R)$. Thus $|\text{ext}(\text{conv}(P \cup A_2 P))| = |\text{ext}(\text{conv}(Q_1 \cup A_{2}Q_2 \cup R))| \leq |Q_1| + |A_{2}Q_2| + |R| \leq ((|Q_1| + |Q_2| + |\{r, s\}|) + 2 \leq |\text{ext}(P)| + 2$. The result for any $P \subseteq \mathbb{R} \times \mathbb{R}_-$ can be proved similarly.

Proposition 11. The pair $\Sigma_2$ has the oligo-vertex property and $N_k(\Sigma_2) = O(k)$. 

17
Proof. To simplify the notation, we omit the dependence of Σ 2 in N k(Σ 2, a) and P k(Σ 2, a) in the rest of this proof. We claim that N k+1(a) ≤ N k(a) + 8 for any a ∈ R 2 and integer k ≥ 2. Then
N k(a) ≤ N k−1(a) + 8 ≤ · · · ≤ N 2(a) + 8(k − 2) ≤ 4 + 8(k − 2) = 8k − 12. Thus N k = O(k).

To prove the claim, first observe that P k+1(a) = conv(A 1P k(a) ∪ A 2A 1P k(a)) = conv(A 1P k(a) ∪ A 2A 1P k(a)). Define P + = A 1P k(a) ∩ {x | x 2 ≥ 0} and P − = A 1P k(a) ∩ {x | x 2 ≤ 0}. Notice that P + is a polytope in R × R_+ and P − is polytope in R × R_−, and |ext(P +)| + |ext(P −)| ≤ |ext(A 1P k(a))| + 4 = N k(a) + 4. The first inequality above follows from the fact the line x 2 = 0 may introduce two new extreme points for both P + and P −. On the other hand,
P k+1(a) = conv(A 1P k(a) ∪ A 2A 1P k(a))
= conv(P + ∪ P − ∪ A 2(P + ∪ P −))
= conv(conv(P + ∪ A 2P +) ∪ conv(P − ∪ A 2P −)).

Thus N k+1(a) ≤ |ext(conv(conv(P + ∪ A 2P +)))| + |ext(conv(P − ∪ A 2P −))|. By Lemma 6, |ext(conv(P + ∪ A 2P +)))| ≤ |ext(P +)| + 2 and |ext(conv(P − ∪ A 2P −))| ≤ |ext(P −)| + 2. Thus we have N k+1(a) ≤ |ext(P +)| + 2 + |ext(P −)| + 2 ≤ N k(a) + 8. □

5.3  Σ 3 = {A 2, A 3}

We first prove the following result when the initial vector a is in the first quadrant.

Proposition 12. For any a ∈ Q 1, N k(Σ 3, a) = O(k).

Proof. The proof is similar to the proof of Proposition 8 for Σ 1. We first bound the cardinality of E k(Σ 3, a) for each i. Similar to the proofs of Lemmas 2, 3, and 4, we can show that for any a ∈ Q 1, |E k(Σ 3, a)| ≤ 4 when k ≥ 3, E k(Σ 3, a) ⊆ {A 2a, A 3a} when k ≥ 1, |E k(Σ 3, a)| ≤ |E k−1(Σ 3, a)| + |E k−1(Σ 3, a)| + 2 and |E k(Σ 3, a)| ≤ |E k−1(Σ 3, a)| + |E k−1(Σ 3, a)| + 2 when k ≥ 1, respectively. Then |E k(Σ 3, a)| ≤ |E k−1(Σ 3, a)| + 6 ≤ |E k(Σ 3, a)| + 6( k − 3) ≤ 6k − 10. Similarly, |E k(Σ 3, a)| ≤ 6k − 10. Finally, for any a ∈ Q 1 and integer k ≥ 3, N k(Σ 3, a) ≤ Σ 10 |E k(Σ 3, a)| ≤ 8 + 4 + (6k − 10) + 2 + (6k − 10) + 8 = 12k − 6. □

Proposition 13. The pair Σ 3 has the oligo-vertex property and N k(Σ 3) = O(k 2).

Proof. We only need to prove that N k(Σ 3, a) = O(k 2) for any a ∈ int(Q 4). Define f k = sup{N k(Σ 3, a) | a ∈ int(Q 4)} for any integer k ≥ 1. Note that f k = sup{N k(Σ 3, a) | a ∈ int(Q 4)} for k ≥ 1 as well. Since P k(Σ 3, a) = conv(P k−1(Σ 3, A 2a) ∪ P k−1(Σ 3, A 3a)), we have N k(Σ 3, a) ≤ N k−1(Σ 3, A 2a) + N k−1(Σ 3, A 3a). Consider a vector a = (a 1, a 2) ⊤ ∈ int(Q 4) with a 1 > 0 and a 2 < 0.
1. If \( a_1 = -a_2 \), we have \( A_2a = (0, a_2) \top \in Q_3 \) and \( A_3a = (a_1, 0) \top \in Q_1 \). Then there exists \( \alpha > 0 \) and integer \( k_0 \) such that for any integer \( l \geq k_0 \), \( N_l(\Sigma_3, A_2a) \leq \alpha l \) and \( N_l(\Sigma_3, A_3a) \leq \alpha l \). Thus for any integer \( k \geq k_0 + 1 \), \( N_k(\Sigma_3, a) \leq N_{k-1}(\Sigma_3, A_2a) + N_{k-1}(\Sigma_3, A_3a) \leq \alpha(k-1) + \alpha(k-1) \leq 2\alpha k_0 \). Therefore, \( N_k(\Sigma_3, a) = O(k) \).

2. If \( a_1 < -a_2 \), we have \( A_2a = (a_1 + a_2, a_2) \top \in Q_3 \) and \( A_3a = (a_1, a_1 + a_2) \top \in \text{int}(Q_4) \). Then there exists \( \alpha > 0 \) and integer \( k_0 \) such that for any integer \( l \geq k_0 \), \( N_l(\Sigma_3, A_2a) \leq \alpha l \) and \( N_l(\Sigma_3, A_3a) \leq \alpha l \). For any \( k \geq k_0 + 1 \), \( N_k(\Sigma_3, a) \leq N_{k-1}(\Sigma_3, A_2a) + N_{k-1}(\Sigma_3, A_3a) \leq \alpha(k-1) + f_{k-1} \). Then for any \( k \geq k_0 + 1 \), \( f_k = \alpha(k-1) + f_{k-1} \). Thus for any \( k \geq 2k_0 \),

\[
\begin{align*}
f_k & \leq \alpha(k-1) + f_{k-1} \\
& \leq \alpha(k-1) + \alpha(k-2) + f_{k-2} \\
& \cdots \leq \alpha(k-1) + \alpha(k-2) + \cdots + \alpha k_0 + f_{k_0} \\
& \leq \alpha \left( \frac{(k-1) + k_0}{2} \right) + 2^{k_0} \leq \beta k^2,
\end{align*}
\]

for some \( \beta > 0 \). Therefore, \( f_k = O(k^2) \).

3. If \( a_1 > -a_2 \), it can be proved that \( f_k = O(k^2) \) with a similar argument as in the case \( a_1 < -a_2 \).

\[\square\]

5.4 \( \Sigma_4 = \{ A_4, A_5 \} \)

**Proposition 14.** The pair \( \Sigma_4 \) has the oligo-vertex property and \( N_k(\Sigma_4) = O(k^2) \).

**Proof.** First observe that \( A_4A_5 = A_2A_2, A_4A_4 = A_2A_3, A_5A_5 = A_3A_2, \) and \( A_5A_4 = A_3A_3 \). When \( k \) is an even integer, every product of \( k \) matrices with \( A_2 \) and \( A_3 \) can be represented by a product of \( k \) matrices with \( A_1 \) and \( A_4 \) and vice versa. Therefore, for any given \( a \in \mathbb{R}^2 \), \( P_k(\Sigma_4, a) = P_k(\Sigma_3, a) \) and \( N_k(\Sigma_4, a) = N_k(\Sigma_3, a) \). When \( k \) is an odd integer, \( P_k(\Sigma_4, a) = \text{conv}(A_4P_{k-1}(\Sigma_4, a) \cup A_5P_{k-1}(\Sigma_4, a)) \) and \( N_k(\Sigma_4, a) \leq 2N_{k-1}(\Sigma_4, a) = 2N_{k-1}(\Sigma_3, a) \). Since there exists \( \alpha > 0 \) and integer \( k_0 \) such that \( N_k(\Sigma_3, a) \leq \alpha k^2 \) for any integer \( k \geq k_0 \), we have \( N_k(\Sigma_4, a) \leq 2\alpha k^2 \) for any integer \( k \geq k_0 \). Therefore, \( N_k(\Sigma_4) = O(k^2) \).

\[\square\]

5.5 \( \Sigma_5 = \{ A_2, A_4 \} \)

**Proposition 15.** For any \( a \in Q_1 \) with \( a_1 \geq a_2 \), \( N_k(\Sigma_5, a) = O(k) \).

**Proof.** First similar to the proofs of Lemmas 2 and 3, we can show by induction that for any \( a \in \mathbb{Q}_1 \) with \( a_1 \geq a_2 \), \( E_k(\Sigma_5, k) = \{ A_k^i a \} \) and \( E_k(\Sigma_5, a) = \{ A_k^i a \} \) when \( k \geq 0 \), and \( |E_k^1(\Sigma_5, a)| \leq |E_k^{i-1}(\Sigma_5, a)| + |E_k^{i-1}(\Sigma_5, a)| + 2 \) and \( |E_k^2(\Sigma_5, a)| \leq |E_k^{i-1}(\Sigma_5, a)| + |E_k^{i-1}(\Sigma_5, a)| + 2 \) when \( k \geq 1 \), respectively. Then \( |E_k^1(\Sigma_5, a)| \leq |E_{k-1}^1(\Sigma_5, a)| + 3 \leq |E_{k-1}^1(\Sigma_5, a)| + 3(k-1) \leq 3k-1 \). Similarly, \( |E_k^2(\Sigma_5, a)| \leq 3k-1 \). Finally, for any integer \( k \geq 1 \), \( N_k(\Sigma_5, a) \leq \sum_{i=0}^4 |E_i^1(\Sigma_5, a)| \leq 6k + 8 \).

We now extend Proposition 15 to the case where \( a \) is in the first quadrant.

**Proposition 16.** For any \( a \in Q_1 \), \( N_k(\Sigma_5, a) = O(k) \).

**Proof.** For any \( a = (a_1, a_2) \) with \( a_1 \geq 0 \) and \( a_2 \geq 0 \), both \( A_2a \) and \( A_4a \) are contained in \( \{ x \in \mathbb{R}_+^2 \mid x_1 \geq x_2 \} \). By Proposition 15, there exists \( \alpha > 0 \) and integer \( k_0 \) such that for any integer \( l \geq k_0 \), \( N_l(\Sigma_5, A_2a) \leq \alpha l \) and \( N_l(\Sigma_5, A_4a) \leq \alpha l \). Thus for any integer \( k \geq k_0 + 1 \), \( N_k(\Sigma_5, a) \leq N_{k-1}(\Sigma_5, A_2a) + N_{k-1}(\Sigma_5, A_4a) \leq \alpha(k-1) + \alpha(k-1) \leq 2\alpha k_0 \). Therefore, \( N_k(\Sigma_5, a) = O(k) \).

Finally, we extend the result to \( a \in \mathbb{R}^2 \), similar to Proposition 13 for the case \( \Sigma_3 \).

**Proposition 17.** The pair \( \Sigma_5 \) has the oligo-vertex property and \( N_k(\Sigma_5) = O(k^2) \).
6 Computational results

In this section, we compare the performance of our algorithm with one state-of-the-art global optimization solver Baron [20]. We randomly generate 10 instances for each of the 10 sets of parameters \((n,m,K)\) for (P), with 100 instances in total. The parameters are summarized in Table 2. The entries of each matrix are randomly drawn from a uniform distribution over \([-1,1]\), and the entries of the initial vector \(a\) are randomly drawn from an uniform distribution over \([0,1]\).

Note that our algorithm does not rely on any additional property of \(f\) other than convexity. In order for Baron to gain a better performance, we choose a simple smooth objective function \(f(x) = \|x\|_2^2\).

The mixed-integer nonlinear programming (MINLP) formulation of (P) is given in (11) and solved by Baron, where \(A_{ij}\) denotes the \((i,j)\)-th entry of the \(l\)-th matrix for \(l \in [m]\). Note that we also tried to linearize the constraints in the MINLP formulation by introducing big-M constants, but we observed that Baron easily run into numerical issues with many big-M constants in the constraints, even for a small-sized instance.

\[
\begin{align*}
\max_{x,z} & \quad \sum_{i=1}^{n} x_i^2(K) \\
\text{s.t.} & \quad x_i(k) = \sum_{l=1}^{m} \sum_{j=1}^{n} A_{ij} x_j(k-1) z_{k,l}, i \in [n], k \in [K], \\
& \quad \sum_{l=1}^{m} z_{k,l} = 1, k \in [K], \\
& \quad z_{k,l} \in \{0,1\}, l \in [m], k \in [K], \\
& \quad x(0) = a.
\end{align*}
\]

(11)

Our algorithm is coded in Matlab. Computational experiments are conducted on a Laptop with Intel i7-6560U 2.20 GHz and 8 GB of RAM memory, under Windows 10 Operating System. The MINLP formulation is coded in AMPL and solved by Baron 18.5.8. The time limit for each instance is set to 600s. When \(n \leq 5\), our algorithm employs Matlab’s build-in function \(convhulln\) to construct the set of extreme points directly. When \(n \geq 6\), our algorithm solves a linear program with the commercial solver Gurobi [13] to identify each extreme point. The computational results are summarized in Table 2. All test instances are solved to optimality by our algorithm within the time limit. The average solution time of our algorithm is reported in the rows “Our algorithm (s)”. On the other hand, Baron cannot solve most instances to optimality, and has a variety of output for instances of different sizes. Instead of reporting the solution time, we report the number of instances with different outputs by Baron in three categories that were described in [27]: The symbol G (G!) denotes that Baron finds a global optimal solution and proves (cannot prove) its optimality within the time limit; The symbol Limit denotes that Baron finds some feasible solution within the time limit; The symbol Wrong denotes that Baron reports infeasibility or failure.

Our proposed algorithm has a clear advantage over Baron in solving (P). Our algorithm is very efficient in solving instances with \(n = 2\) and large \(m\) and \(K\), requiring less than one second. When \(n\) increases to 8 and 10, our algorithm is able to solve instances with \(K = 50\) and \(K = 20\) respectively in less than one minute. On the other hand, Baron is only able to solve several instances with a pair of \(2 \times 2\) matrices to optimality. When \(n\) or \(m\) is larger than 2, it either cannot find the optimal solution within the time limit or runs into numerical issues. Finally, we observe that when the problem dimension \(n \geq 8\), our algorithm is not able to solve instances with \(K = 100\) within the time limit, since the running time grows rapidly with \(K\). We suspect the reason to be that
the set of randomly generated matrices no longer has the oligo-vertex property for larger $n$. This observation is also consistent with the fact that (P) is NP-hard for general $n$.

## 7 Open Problems and Conclusions

The problem (P) has many applications in operations research and control, and can also be seen as an approximation to the dynamics of more general continuous-time nonlinear switched systems. In this paper, we present an efficient exact algorithm to solve large-sized instances of (P) that cannot be handled by state-of-the-art optimization software. We introduce an interesting property—the oligo-vertex property—for a set of matrices. We now present several open questions on the oligo-vertex property, which we believe may be of independent interest.

1. Does any pair of $2 \times 2$ real matrices have the oligo-vertex property?

2. Is there an “easy-to-check” necessary condition for a set of matrices to have the oligo-vertex property? Is there a finite-time algorithm to test the oligo-vertex property for a given set of matrices with rational entries? If so, is deciding whether a set of matrices has the oligo-vertex property in P or NP?

3. Does the finiteness property imply the oligo-vertex property, and vice versa?

4. Is $N_k(\Sigma) = O(k)$ for any pair of $2 \times 2$ binary matrices?

The last question comes from our observation that $N_k(\Sigma, a)$ grows linearly with $k$ for any $2 \times 2$ binary matrices in the computational experiment. We believe answers to any of the above questions will lead to a faster exact algorithm for (P).

## References


