Optimal switching sequence for switched linear systems

Qie He, **Presenting author**, Zeyang Wu

University of Minnesota, Department of Industrial and Systems Engineering

A discrete optimization problem

We focus on the following discrete-time switched linear system:

$$x(k+1) = T_k x(k), \qquad T_k \in \Sigma,\ k = 0, 1, \dots, \quad (1)$$

where the initial vector $x(0)$ is a given n-dimensional real vector a and the set Σ contains m $n \times n$ real matrices, each of which describes the dynamics of a linear subsystem.

We aim to find a sequence of K matrices, each chosen from Σ, to maximize a convex function over $x(K)$. In particular, we are interested in the following optimization problem (P):

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

An example of switched systems

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.

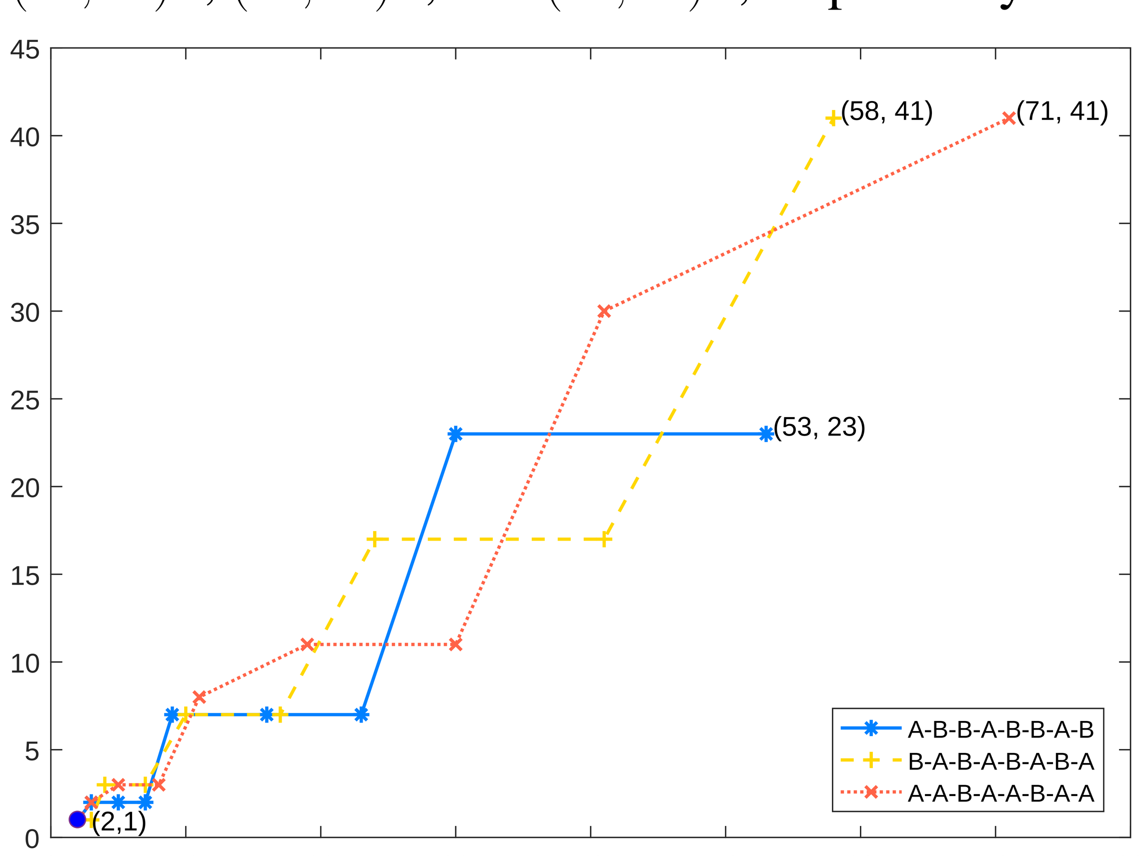

Figure 1: The trajectory of $x(k)$ under different matrix sequences

Applications and main results

Problem (P) has many practical applications and is closely connected to several fundamental problems in control and optimization.

- Applications: Mitigating antibiotic resistance, matrix K-mortality problem, computing the joint spectral radius of a set of matrices, etc.

Our contributions.

- We show that this problem is NP-hard for a pair of stochastic matrices or binary matrices.
- We propose a polynomial-time exact algorithm for the problem when all input data are rational and the given set of matrices Σ has the oligo-vertex property, a new concept we introduce below.

The oligo-vertex property

Let $P_k(\Sigma, a)$ be the convex hull of all possible values of $x(k)$, i.e.,

$$P_k(\Sigma, a) := \mathrm{conv}(\{x(k) \mid x(k) = T_{k-1} \cdots T_0 a,\ T_j \in \Sigma, j = 0, \dots, k-1\}).$$

Let $N_k(\Sigma, a)$ be the number of extreme points of $P_k(\Sigma, a)$ and $N_k(\Sigma) = \sup_{a \in \mathbb{R}^n}\{N_k(\Sigma, a)\}$.

Definition. A set of matrices Σ is said to have the **oligo-vertex property** if there exists some constant d such that $N_k(\Sigma) = O(k^d)$.

The oligo-vertex property indicates that the number of extreme points of $P_k(\Sigma, a)$ grows at most polynomially in k regardless of the initial vector a, although the number of possible values of $x(k)$ grows exponentially with k in general.

Sufficient conditions for a set of matrices to have the oligo-vertex property:

- A finite set of matrices that commute;
- A finite set of matrices containing at most one matrix with the rank higher than one;
- A pair of 2×2 matrices sharing at least one common eigenvector;
- A pair of 2×2 binary matrices.

We also show that the oligo-vertex property is invariant under a similarity transformation.

Examples

Σ_1 contains a pair of 2×2 binary matrices. Σ_1 has the oligo-vertex property.

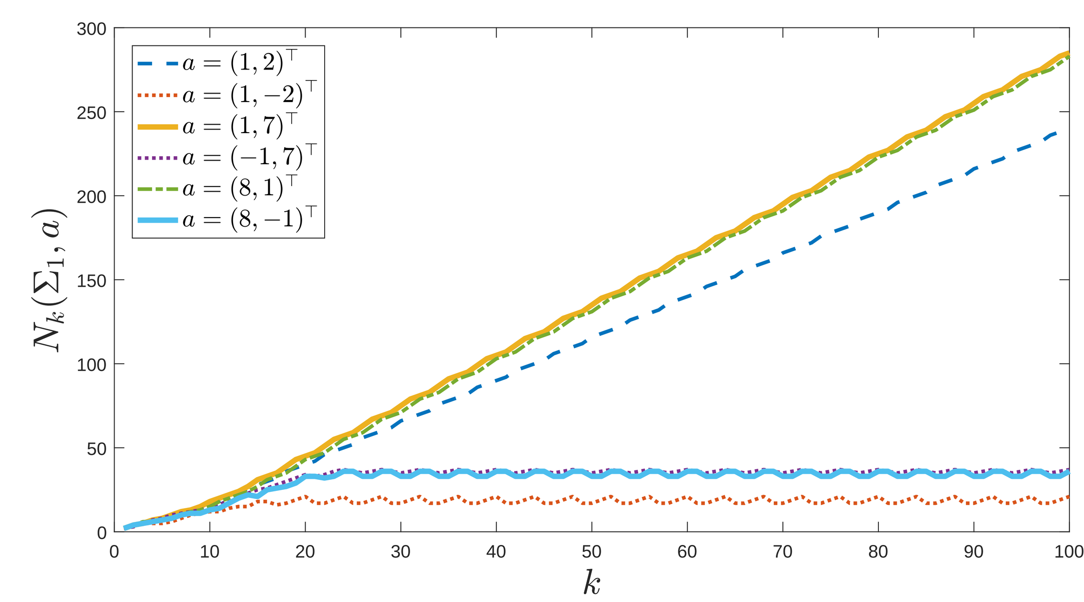

Figure 2: $N_k(\Sigma, a)$ with Σ_1 under different initial a's

Σ_2 contains five 2×2 matrices whose entries are randomly drawn from an uniform distribution on $[0, 1]$. Σ_2 is likely to have the oligo-vertex property since the dimension $n = 2$.

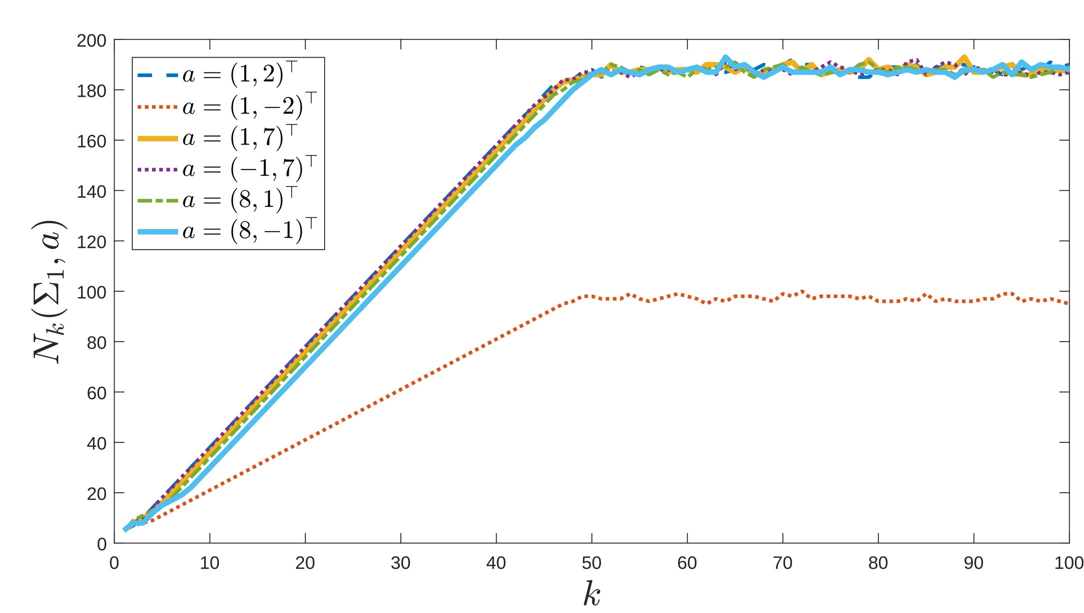

Figure 3: $N_k(\Sigma, a)$ with $|\Sigma| = 5$ under different initial a's

Σ_3 contains two 5×5 matrices whose entries are randomly drawn from an uniform distribution on $[0, 1]$. Σ_3 is not likely to have the oligo-vertex property.

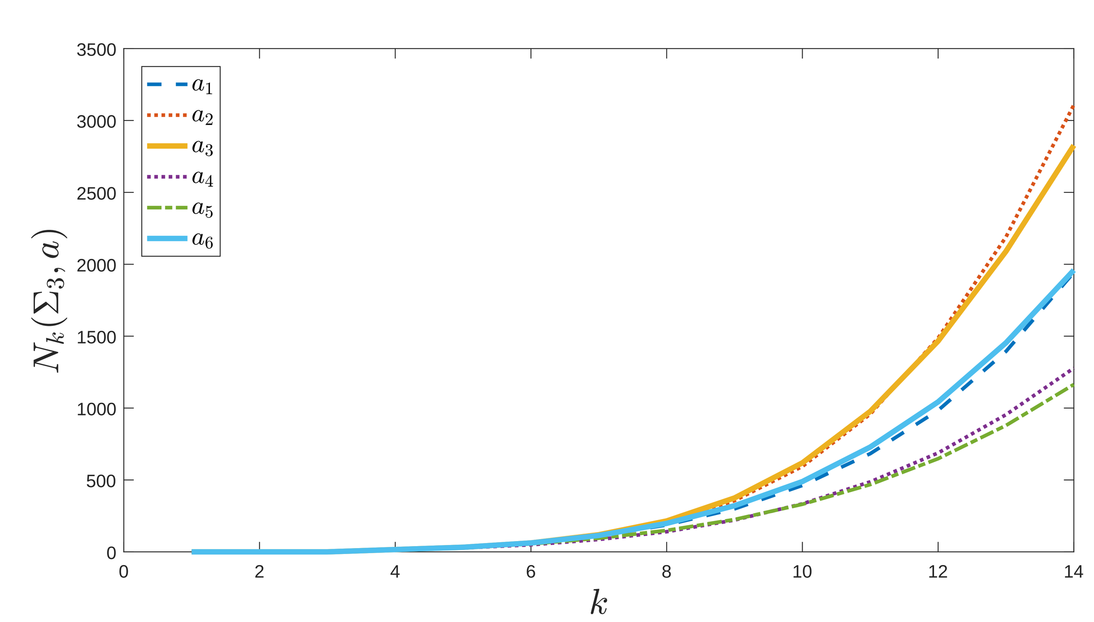

Figure 4: $N_k(\Sigma, a)$ with $n = 5$ under different initial a's

The algorithm

In this section, we present an exact algorithm to solve (P). The algorithm is described in Algorithm 1. A key step of Algorithm 1 is to construct $P_k(\Sigma, a)$ sequentially for $k = 0, 1, \dots, K$.

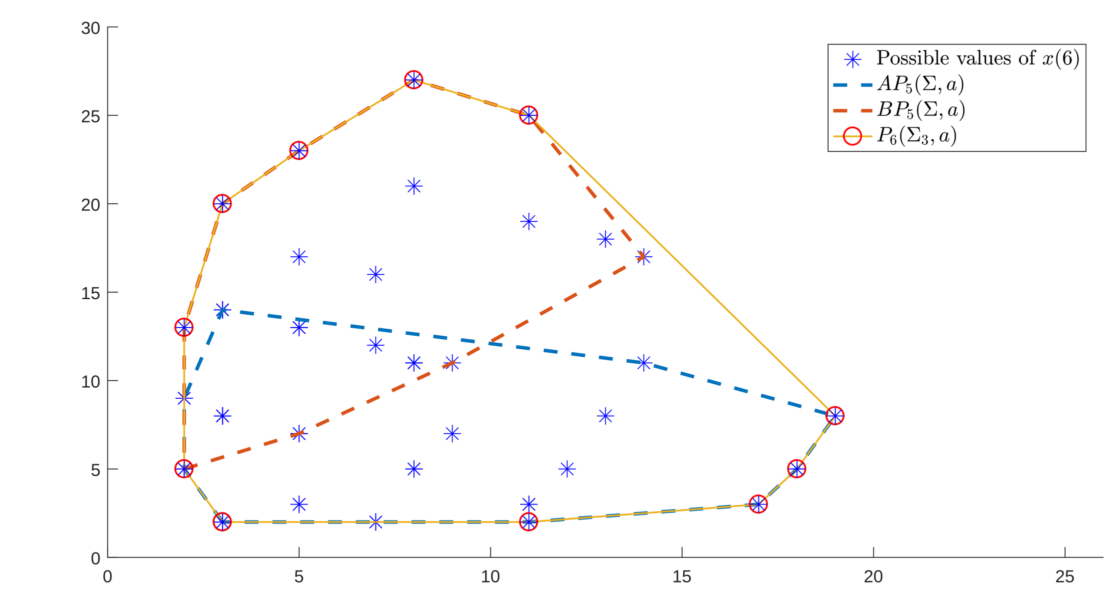

Figure 5: An example of $P_6(\Sigma, a)$ with $\Sigma = \{A, B\}$, $P_6(\Sigma, a) = \mathrm{conv}(AP_5(\Sigma, a) \cup BP_5(\Sigma, a))$

Assume that $\Sigma = \{A_1, \dots, A_m\} \in \mathbb{Z}^{n \times n}$ and $a \in \mathbb{Z}^n$.

Algorithm 1 An algorithm to solve (P).

1: **Initialize:** Set $E_0 = \{a\}$.
2: **for** $k = 0, 1, \dots, K-1$ **do**
3: $\quad F_k^i = A_i E_k$ for $i = 1, \dots, m$.
4: $\quad$ Construct E_{k+1} to be the set of extreme points of $\mathrm{conv}(\cup_{i=1}^m F_k^i)$.
5: **end for**
6: Find an $x^*(K) \in \arg\max\{f(x) \mid x \in E_K\}$ by enumeration.
7: Retrieve the optimal matrix sequence $T_{K-1}, T_{K-2}, \dots, T_0$ from $x^*(K)$.

Complexity result. Suppose $N_k(\Sigma) = O(k^d)$, the dimension is n, $|\Sigma| = m$, M is the largest magnitude of all input data.

- When the linear programming approach is employed to construct E_{k+1} at Step 4, then (P) can be solved in $O(m^2 n^{4.5} K^{2d+2}(\log n + \log M))$ time.
- When $n = 2$ and Graham's scan is employed at Step 4 of Algorithm 1, then (P) can be solved in $O(mK^{d+1}(\log m + \log K))$ time.
- In particular, we conjecture that when Σ is a pair of 2×2 matrices, (P) can be solved in $O(K^2 \log K)$ time.