Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A matrix-based static approach to analysis of finite state machines*

He DENG[1], Yongyi YAN[†‡1], Zengqiang CHEN[2]

[1]College of Information Engineering, Henan University of Science and Technology, Luoyang 471000, China

[2]College of Artificial Intelligence, Nankai University, Tianjin 300071,China

[†]E-mail: yyyan@mail.nankai.edu.cn

Received Dec. 3, 2021; Revision accepted Feb. 23, 2022; Crosschecked May 6, 2022

Abstract: Traditional matrix-based approaches in the field of finite state machines construct state transition matrices, and then use the powers of the state transition matrices to represent corresponding dynamic transition processes, which are cornerstones of system analysis. In this study, we propose a static matrix-based approach that revisits a finite state machine from its structure rather than its dynamic transition process, thus avoiding the "explosion of complexity" problem inherent in the existing approaches. Based on the static approach, we reexamine the issues of closed-loop detection and controllability for deterministic finite state machines. In addition, we propose controllable equivalent form and minimal controllable equivalent form concepts and give corresponding algorithms.

Key words: Logical systems; Finite-valued systems; Semi-tensor product of matrices; Finite state machines; Matrix approaches

https://doi.org/10.1631/FITEE.2100561 **CLC number:** TP13

1 Introduction

Matrix-based approaches have a wide range of applications in the field of finite state machines (Lu et al., 2018; Chen et al., 2020). For matrix-based approaches, there are two major mathematical model types: the state transition matrix model (TM model) (Xu XR and Hong, 2013a; Chen et al., 2020), based on the conventional matrix product, and the matrix model based on the semi-tensor product (STP) model (Xu XR and Hong, 2013b; Zhu R et al., 2022). Several representative results are presented (Lu et al., 2017; Yan et al., 2022). For a finite state machine, the TM model is effective for closed-ended issues such as controllability and reachability. However, the computational complexity of the TM model is often exponential for the optimal path issue or for finding relevant inputs. The major reason is that some input information is lost in modeling formalism. Fortunately, the missing information can be added to the TM model with the help of STP theory, creating the STP model (Xu XR and Hong, 2013a; Han et al., 2018). In general, the STP model contains all information on a dynamic process, both information of state transitions and that of inputs. Thus, the STP model can solve all issues in the field of finite state machines theoretically. However, a drawback of the STP model is the "explosion of dimension" problem (Yue et al., 2019; Yan et al., 2021); that is, the dimension of state transition matrices in the STP model increases exponentially with increase in the time step. This problem also occurs in the TM model, where the complexity increases polynomially as the time step increases linearly (Xu Q et al., 2021). The "explosion of complexity" is caused by the repeated product of state transition matrices (Cheng

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. U1804150, 62073124, and 61973175)

ORCID: He DENG, https://orcid.org/0000-0002-9646-578X; Yongyi YAN, https://orcid.org/0000-0002-7181-5894

and Qi, 2010; Yan et al., 2014). One of the major reasons for the repeated product is that state transition processes are always considered as dynamic processes, which implies that each state transition requires a product of the state transition matrix.

Motivated by the "explosion of complexity" inherent in the existing approaches, in contrast to traditional matrix-based approaches, we propose a static matrix-based approach, which relies on the structure of a finite state machine itself rather than its dynamic process, thus avoiding the "explosion of complexity." Based on the static approach, we reexamine the issue of closed-loop detection in deterministic finite state machines. By the fact that all states in a closed loop are mutually controllable, we present the definitions and algorithms of a controllable equivalent form and a minimal controllable equivalent form. With a static view of the closed loop, the two-state controllability issue can be resolved only once, by matrix division, making use of our algorithms in the best case.

2 Preliminaries

Notations: $\boldsymbol{M}_{m\times n}$ is the set of $m \times n$ real matrices; $\mathrm{col}_i(\boldsymbol{A})$ is the i^{th} column of matrix $\boldsymbol{A}$; $\boldsymbol{\delta}_n^i$ is the i^{th} column of the identity matrix of dimension n; $|C|$ is the cardinality of a finite set C; $\mathbf{1}_m = \underbrace{[1, 1, \cdots, 1]}_{m}^{\mathrm{T}}$; $A(i, j)$ is the entry for a matrix $\boldsymbol{A}$ in row i and column j; $\varDelta_n := \{\boldsymbol{\delta}_n^1, \boldsymbol{\delta}_n^2, \cdots, \boldsymbol{\delta}_n^n\}$; $\wedge$ and $\vee$ are the logical operators of "And" and "Or," respectively; $\mathrm{ind}(a)$ is the in-degree of a state a; $\boldsymbol{\delta}_n\{i_1, i_2, \ldots, i_n\} := \left\{\boldsymbol{\delta}_n^{i_1}, \boldsymbol{\delta}_n^{i_2}, \ldots, \boldsymbol{\delta}_n^{i_n}\right\}$.

We assume that the reader is familiar with the basic notions and concepts of finite state machines and STP. Identify x_i with $\boldsymbol{\delta}_n^i$ (or $\boldsymbol{x}_i$) $(1 \le i \le n)$, expressed as $x_i \sim \boldsymbol{\delta}_n^i(\boldsymbol{x}_i)$ for simplicity, and call $\boldsymbol{\delta}_n^i(\boldsymbol{x}_i)$ the vector form of x_i. In the framework of matrix-based approaches, the dynamics of $A = (X, E, f, x_0)$ with input $e = e_1e_2\ldots e_t \in E^*$ can be formulated as follows:

1. STP model:

$$\boldsymbol{x}(t) = \tilde{\boldsymbol{F}}^t \ltimes \boldsymbol{\delta}_n^{x_0} \ltimes \boldsymbol{u}(t), \tag{1}$$

where $\boldsymbol{u}(t) = \ltimes_{i=1}^t \boldsymbol{\delta}_m^i = \boldsymbol{\delta}_m^1 \ltimes \boldsymbol{\delta}_m^2 \ltimes \ldots \ltimes \boldsymbol{\delta}_m^t$, $\boldsymbol{\delta}_m^j$ is the vector form of e_j, $j = 1, 2, \ldots, t$, $\tilde{\boldsymbol{F}} = \boldsymbol{F} \ltimes \boldsymbol{W}_{[n,m]}$, $\boldsymbol{W}_{[n,m]}$ is a swap matrix (Zhu SM et al., 2021), and $\boldsymbol{F} = [\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_m]$ is defined as

$$F_i(s,t) = \begin{cases} 1, \text{ if } \boldsymbol{\delta}_n^s \in f(\boldsymbol{\delta}_n^t, \boldsymbol{\delta}_m^i), \\ 0, \text{ otherwise.} \end{cases} \tag{2}$$

2. TM model:

$$\boldsymbol{x}(t) = \boldsymbol{T}_{e_t} \cdots \boldsymbol{T}_{e_2} \boldsymbol{T}_{e_1} \boldsymbol{\delta}_n^{x_0}. \tag{3}$$

Eqs. (1) and (3) are standard matrix models in that they construct the state transition matrices, $\tilde{\boldsymbol{F}}$ in the STP model and $\boldsymbol{T}_{e_i}$ $(1 \le i \le t)$ in the TM model, to describe the dynamics of $\boldsymbol{A}$. Note that we are more interested in the outcome of a state transition than in the inputs that can cause the transition. Therefore, we modify Eq. (3) as

$$\boldsymbol{x}(t) = \boldsymbol{T}^t \boldsymbol{\delta}_n^{x_0}, \tag{4}$$

where $\boldsymbol{T} = \sum\limits_{j=1}^{t} \boldsymbol{T}_{e_j}$.

We simplify notations by writing $\boldsymbol{\delta}_n^X$ instead of $\sum\limits_{x_i\in X} \boldsymbol{\delta}_n^i$ and $\mathrm{ind}(X)$ instead of $\sum\limits_{x\in X} \mathrm{ind}(x)$. $\varPsi(\boldsymbol{\eta}) := \{\boldsymbol{\delta}_n^k | \text{the } k^{\mathrm{th}} \text{ element of } \boldsymbol{\eta} \text{ is non-zero for any } \boldsymbol{\eta} \in \boldsymbol{M}_{n\times 1}\}$. For example, let $\boldsymbol{M} = [1\ \ 0\ \ 1\ \ 0]^{\mathrm{T}}$. Then we have $\varPsi(\boldsymbol{M}) = \{\boldsymbol{\delta}_4^1, \boldsymbol{\delta}_4^3\}$. Also, we use $\boldsymbol{\delta}_n[i_1, i_2, \ldots, i_n]$ instead of $[\boldsymbol{\delta}_n^{i_1}, \boldsymbol{\delta}_n^{i_2}, \cdots, \boldsymbol{\delta}_n^{i_n}]$ for brevity.

3 Main results

3.1 Static approach for detection of closed loops

The closed loops, the core point of this paper, are proposed in this subsection.

Definition 1 For a deterministic finite state machine (DFSM) $M = (X = \varDelta_n, E, f, x_0)$, a state set $X_{\mathrm{s}} \subseteq X$ is called a single closed loop, if for all $x \in X_{\mathrm{s}}$,

$$(\boldsymbol{T}^{|X_{\mathrm{s}}|}\boldsymbol{\delta}_n^x) .* \boldsymbol{\delta}_n^{X_{\mathrm{s}}} = \boldsymbol{\delta}_n^x,$$

where "$.*$" denotes element-wise multiplication.

Although Definition 1 is an accurate description, it is not less complex than the repeated matrix products in Eqs. (1) and (3). Thus, we give the following definition of a weak version:

Definition 2 For a DFSM $M = (X = \varDelta_n, E, f, x_0)$, a state set $X_{\mathrm{s}} \subseteq X$ is called a single closed loop, if

$$(\boldsymbol{T}\boldsymbol{\delta}_n^{X_{\mathrm{s}}}) .* \boldsymbol{\delta}_n^{X_{\mathrm{s}}} = \boldsymbol{\delta}_n^{X_{\mathrm{s}}} \wedge (\boldsymbol{T}^{\mathrm{T}}\boldsymbol{\delta}_n^{X_{\mathrm{s}}}) .* \boldsymbol{\delta}_n^{X_{\mathrm{s}}} = \boldsymbol{\delta}_n^{X_{\mathrm{s}}}.$$

Note that Definition 2 is not strictly a sufficient condition, but it can be applied to the results presented in this paper and has time complexity $O(1)$.

Definition 3 For a DFSM $M = (X = \varDelta_n, E, f, x_0)$, a state set $X_\mathrm{s} \subseteq X$ is called a compound closed loop, if $((\boldsymbol{T}-\boldsymbol{T}^\mathrm{T})\boldsymbol{\delta}_n^{X_\mathrm{s}}).*\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{0}$, and a state x_i is called a bifurcation state if $\left|\varPsi(\boldsymbol{T}\delta_n^i)\right| > 1$. It is clear that a bifurcation state is the exit for the transitions from states inside a closed loop in which the bifurcation state stays to states outside the closed loop. If a bifurcation state stays in a closed loop, then we say that the bifurcation state forms the closed loop. We use symbol α to refer to a bifurcation state. Also, we use W_A to denote all the single closed loops in DFSM A. Due to the fact that the increase of complexity caused by the increase of the number of bifurcation states is not linear, we give the following function to differentiate DFSMs with different numbers of bifurcation states:

Given a matrix $\boldsymbol{M} \in \boldsymbol{M}_{m\times n}$, $\|\boldsymbol{M}\|$ is defined as $\|\boldsymbol{M}\| = |\{i\,||\varPsi(\mathrm{col}_i(\boldsymbol{M}))| > 1\}|$.

Lemma 1 For any DFSM $M = (X = \varDelta_n, E, f, x_0)$ at $\|\boldsymbol{T}\| = 0$, a state set $X_\mathrm{s} \subseteq X$ is a single closed loop if and only if

$$(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{0} \ \wedge\ (\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X^*{}_\mathrm{s}} \neq \boldsymbol{0},$$

where $X^*{}_\mathrm{s} \in 2^{X_\mathrm{s}}$.

Proof (Necessity) The reachable state set of a single closed loop is itself at $\|\boldsymbol{T}\| = 0$; that is, $\boldsymbol{T}\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{\delta}_n^{X_\mathrm{s}}$ and/or $\boldsymbol{T}^\mathrm{T}\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{\delta}_n^{X_\mathrm{s}}$.

(Sufficiency) We first prove that if $(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{0}\wedge(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X^*{}_\mathrm{s}} \neq \boldsymbol{0}$, then $\boldsymbol{T}\boldsymbol{\delta}_n^x = \boldsymbol{\delta}_n^x$ holds for all $x \in X_\mathrm{s}$. It is easy to see, from the STP properties, that $(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} = (\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{\{x_{i_1},\cdots,x_{i_p}\}} = (\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{x_{i_1}} + \cdots + (\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{x_{i_p}} = \boldsymbol{0}$ $(1 \le p \le n)$. By contradiction, assume that $x_i \in X_\mathrm{s}$ has $\boldsymbol{T}\boldsymbol{\delta}_n^{x_i} \neq \boldsymbol{\delta}_n^{x_i}$ $(i = a, b)$. Then either $(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{x_a} + \cdots + (\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{x_b}\cdots \neq \boldsymbol{0}$ if $(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{x_a} \neq -(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{x_b}$, or $(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{x} = \boldsymbol{0}$ has another basic solution $\boldsymbol{x} = \boldsymbol{\delta}_n^{\{x_a,x_b\}} \in 2^{X_\mathrm{s}}$ if $(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{x_a} = -(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{x_b}$. Hence, a contradiction exists. Next, we prove that for all $x \in X_\mathrm{s}$, $(\boldsymbol{T}^{|X_\mathrm{s}|}\boldsymbol{\delta}_n^x).*\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{\delta}_n^x$ is satisfied. Because $\boldsymbol{T}\boldsymbol{\delta}_n^x = \boldsymbol{\delta}_n^x$ $(x \in X_\mathrm{s})$ holds, we have that $(\boldsymbol{T}^{|X_\mathrm{s}|}\boldsymbol{\delta}_n^x).*\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{\delta}_n^x.*\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{\delta}_n^x$ holds for all $x \in X_\mathrm{s}$.

Algorithm 1 is designed to find all closed loops in a DFSM at $\|\boldsymbol{T}\| = 0$. Note that the solutions of matrix equations are basic solutions in this paper.

Algorithm 1 Finding closed loops in a DFSM at $\|\boldsymbol{T}\| = 0$

1: Construct $\boldsymbol{T}$.
2: The set of all closed loops is $C = \{\varPsi(\boldsymbol{x})|(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{x} = \boldsymbol{0}\wedge(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{x}^* \neq \boldsymbol{0}\}$, where $x^* \in 2^x$.

Next, we consider more general cases and begin with $\|\boldsymbol{T}\| = 1$.

Lemma 2 For any DFSM $M = (X = \varDelta_n, E, f, x_0)$ at $\|\boldsymbol{T}\| = 1$, if a state set $X_\mathrm{s} \subseteq X$ is a single closed loop, then

$$(\tilde{\boldsymbol{T}}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{T}\boldsymbol{\delta}_n^\alpha \vee (\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{0},$$

where $\alpha \in S = \{i\,||\varPsi(\mathrm{col}_i(\boldsymbol{T}))| > 1\}$, and $\tilde{\boldsymbol{T}}$ is defined as

$$\tilde{T}(i,j) = \begin{cases} T(i,j)+1, \text{ if } \boldsymbol{\delta}_n^i = \boldsymbol{\delta}_n^j \in \varPsi(\boldsymbol{T}\boldsymbol{\delta}_n^\alpha), \\ T(i,j), \text{ otherwise.} \end{cases}$$

Proof If X_s is the same type of closed loop as in Lemma 1, then $(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{0}$ is satisfied. Therefore, we assume $(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} \neq \boldsymbol{0}$, which implies that the bifurcation state α must be in X_s. It is easy to find that $\boldsymbol{T}\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{\delta}_n^{X_\mathrm{s}} + \boldsymbol{\delta}_n^{X_\alpha}$, where $\boldsymbol{\delta}_n^{X_\alpha} = \varPsi(\boldsymbol{T}\boldsymbol{\delta}_n^\alpha)\backslash\varPsi(\boldsymbol{\delta}_n^{X_\mathrm{s}})$. X_β is defined as $\boldsymbol{\delta}_n^{X_\beta} = \varPsi(\boldsymbol{T}\boldsymbol{\delta}_n^\alpha)\cap\varPsi(\boldsymbol{\delta}_n^{X_\mathrm{s}})$, and we have $\boldsymbol{\delta}_n^{X_\alpha} + \boldsymbol{\delta}_n^{X_\beta} = \boldsymbol{T}\boldsymbol{\delta}_n^\alpha$. The known conditions are $\boldsymbol{T}$ and α. For X_s, we have $\mathrm{ind}(X_\alpha) = 1$. Because $\boldsymbol{\delta}_n^{X_\alpha} \cap \boldsymbol{\delta}_n^{X_\mathrm{s}} = \varnothing$ and $\boldsymbol{\delta}_n^{X_\beta} \cap \boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{\delta}_n^{X_\beta}$, by making $\tilde{f}(i \in f(\alpha)) := f(i) \cup i$, we have $\mathrm{ind}(X_\mathrm{s}\backslash X_\beta) = 1$, $\mathrm{ind}(X_\alpha) = 1$, and $\mathrm{ind}(X_\beta) = 2$ for X_s. In other words, we have a new state transition matrix $\tilde{\boldsymbol{T}}$ such that the only difference between the two state transition matrices is that the new matrix makes $\tilde{f}(X_\mathrm{s}) - X_\mathrm{s} = X_\alpha + X_\beta$ hold. Then $\tilde{\boldsymbol{T}}\boldsymbol{\delta}_n^{X_\mathrm{s}} - \boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{\delta}_n^{X_\alpha} + \boldsymbol{\delta}_n^{X_\beta} = \boldsymbol{T}\boldsymbol{\delta}_n^\alpha$ is obtained using the vector form.

Lemma 2 is also a necessary condition. To find all single closed loops at $\|\boldsymbol{T}\| = 1$, the idea is to first find all possible single closed loops by the necessary condition, and then to use Definition 2 to determine which ones are single closed loops. From Lemmas 1 and 2, one way to find single closed loops is given in Algorithm 2. Note that Algorithm 2 cannot find compound closed loops. The algorithm for solving compound closed loops is stated later.

Lemma 3 For any DFSM $M = (X = \varDelta_n, E, f, x_0)$ at $\|\boldsymbol{T}\| = 2$, if a state set $X_\mathrm{s} \subseteq X$ is a single closed loop, then

$$\begin{aligned}(\tilde{\boldsymbol{T}}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} &= \boldsymbol{T}\boldsymbol{\delta}_n^S \vee (\hat{\boldsymbol{T}}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} \\ &= \boldsymbol{T}\boldsymbol{\delta}_n^S \vee (\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_\mathrm{s}} = \boldsymbol{0},\end{aligned}$$

where $S=\{i\,|\,|\Psi(\mathrm{col}_i(\boldsymbol{T}))|>1\}$, $\tilde{\boldsymbol{T}}$ is described in Lemma 2, and $\hat{\boldsymbol{T}}$ is defined as

$$\begin{cases}\hat{T}(i,j)=T(i,j)+1, \text{ if } \boldsymbol{\delta}_n^i=\boldsymbol{\delta}_n^j\in\Psi(\boldsymbol{T}\boldsymbol{\delta}_n^{\alpha}),\\ \hat{T}(i,\alpha)=T(i,\alpha)+1, \text{ if } \boldsymbol{\delta}_n^i\in\Psi(\boldsymbol{T}\boldsymbol{\delta}_n^{S\backslash\alpha}),\\ \hat{T}(i,j)=T(i,j), \text{ otherwise,}\end{cases}$$

where $\alpha\in S$.

Algorithm 2 Finding all single closed loops for $M=(X=\Delta_n,E,f,x_0)$ at $\|\boldsymbol{T}\|=1$

Construct $\tilde{\boldsymbol{T}}$ by computing α.
Compute $X_{\mathrm{s}}=\{\boldsymbol{x}|(\tilde{\boldsymbol{T}}-\boldsymbol{I}_{n\times n})\boldsymbol{x}=\boldsymbol{T}\boldsymbol{\delta}_n^{\alpha}\}$.
Apply Definition 2 to X_{s}.
Apply Algorithm 1 to M.

Proof For clarity, suppose that there is only one single closed loop X_{s} and $(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_{\mathrm{s}}}\neq\boldsymbol{0}$. Then we can have only two cases, that is, $S\subset X_{\mathrm{s}}$ (case 1) or $S\not\subset X_{\mathrm{s}}$ (case 2). Assume that $\alpha\in X_{\mathrm{s}}$ and $S\backslash\alpha=\beta\in X_{\mathrm{s}}$ if case 1 occurs, and that $\alpha\in X_{\mathrm{s}}$ and $\beta\notin X_{\mathrm{s}}$ if case 2 occurs. It is easy to find that $\boldsymbol{T}\boldsymbol{\delta}_n^{X_{\mathrm{s}}}-\boldsymbol{\delta}_n^{X_{\mathrm{s}}}=\boldsymbol{\delta}_n^{X_\alpha}+\boldsymbol{\delta}_n^{X_\beta}$ in case 1 where $\boldsymbol{\delta}_n^{X_\alpha}=\Psi(\boldsymbol{T}\boldsymbol{\delta}_n^{\alpha})\backslash\Psi(\boldsymbol{\delta}_n^{X_{\mathrm{s}}})$ and $\boldsymbol{\delta}_n^{X_\beta}=\Psi(\boldsymbol{T}\boldsymbol{\delta}_n^{\beta})\backslash\Psi(\boldsymbol{\delta}_n^{X_{\mathrm{s}}})$, and that $\boldsymbol{T}\boldsymbol{\delta}_n^{X_{\mathrm{s}}}-\boldsymbol{\delta}_n^{X_{\mathrm{s}}}=\boldsymbol{\delta}_n^{X_\alpha}$ in case 2. Case 2 actually involves the same type of closed loops as in Lemma 2. Thus, we take $\tilde{f}(i\in f(\alpha)=X_\alpha\cup X_\gamma):=f(i)\cup i$ to obtain $\tilde{\boldsymbol{T}}\boldsymbol{\delta}_n^{X_{\mathrm{s}}}-\boldsymbol{\delta}_n^{X_{\mathrm{s}}}=\boldsymbol{T}\boldsymbol{\delta}_n^{\alpha}+\boldsymbol{\delta}_n^{\beta}=\boldsymbol{T}\boldsymbol{\delta}_n^{S}$ in case 1 and $\tilde{\boldsymbol{T}}\boldsymbol{\delta}_n^{X_{\mathrm{s}}}-\boldsymbol{\delta}_n^{X_{\mathrm{s}}}=\boldsymbol{T}\boldsymbol{\delta}_n^{\alpha}$ in case 2. Note that although S is known, α and X_β are unknown. To obtain $\boldsymbol{T}\boldsymbol{\delta}_n^{S}$, we make $\hat{f}(\alpha):=f(\alpha)\cup X_\beta$. Then case 2 becomes $\hat{\boldsymbol{T}}\boldsymbol{\delta}_n^{X_{\mathrm{s}}}-\boldsymbol{\delta}_n^{X_{\mathrm{s}}}=\boldsymbol{T}\boldsymbol{\delta}_n^{\alpha}+\boldsymbol{T}\boldsymbol{\delta}_n^{\beta}=\boldsymbol{T}\boldsymbol{\delta}_n^{S}$. As a consequence, cases 1 and 2 both have the form $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$. Finally, for the situation where two single closed loops occur, the same result $(\tilde{\boldsymbol{T}}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_{\mathrm{s}}}=\boldsymbol{T}\boldsymbol{\delta}_n^{S}$ can be obtained from Lemma 2.

It is clear that a single closed loop can contain more than one bifurcation state. Thus, for a DFSM A, we use θ to denote all bifurcation states that can form a closed loop in A, and θ_α for all bifurcation states of the closed loop in which α is a bifurcation state, that is, $\theta_\alpha:=\left\{\boldsymbol{\delta}_n^i\,|\exists y\in W_A,\ x_i,\alpha\in y\wedge i,j\in S\right\}$ $\left(S=\{i\,||\Psi(\mathrm{col}_i(\boldsymbol{T}))|>1\},\boldsymbol{\delta}_n^j\sim\alpha\right)$.

Lemma 4 For any DFSM $M=(X=\Delta_n,E,f,x_0)$ at $\|\boldsymbol{T}\|>2$, if a state set $X_{\mathrm{s}}\subseteq X$ is a single closed loop, then

$$(\widehat{\boldsymbol{T}}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_{\mathrm{s}}}=\boldsymbol{T}\boldsymbol{\delta}_n^{S}\vee(\boldsymbol{T}-\boldsymbol{I}_{n\times n})\boldsymbol{\delta}_n^{X_{\mathrm{s}}}=\boldsymbol{0},$$

where $\widehat{\boldsymbol{T}}$ is defined as

$$\begin{cases}\widehat{T}(i,j)=T(i,j)+1, \text{ if } \boldsymbol{\delta}_n^i=\boldsymbol{\delta}_n^j\in\Psi(T\boldsymbol{\delta}_n^{\theta_\alpha}),\\ \widehat{T}(i,\alpha)=T(i,\alpha)+1, \text{ if } \boldsymbol{\delta}_n^i\in\Psi(T\boldsymbol{\delta}_n^{S\backslash\theta_\alpha}),\\ \widehat{T}(i,j)=T(i,j), \text{ otherwise,}\end{cases}$$

where $\alpha\in\theta_\alpha$.

Proof Lemma 4 is obtained by arbitrarily combining cases 1 and 2 in Lemma 3.

Corollary 1 Given a DFSM $M=(X=\Delta_n,E,f,x_0)$, $S=\{i\,||\Psi(\mathrm{col}_i(\boldsymbol{T}))|>1\}$ can be divided into $H=\{h\,|\boldsymbol{x}_h\in\boldsymbol{\delta}_nS\wedge x_h\in\theta_\alpha\}$ and $L=\{l\,|\boldsymbol{x}_l\in\boldsymbol{\delta}_nS\wedge x_l\notin\theta_\alpha\}$ for any $\alpha\in S$.

Note that the key to Lemma 4 is θ_α. Given a bifurcation state α that may form a single closed loop, one way to find θ_α is given in Algorithm 3. Algorithm 3 has time complexity $O(\sum\limits_{s\in S\backslash\alpha}|\Psi(\boldsymbol{T}\boldsymbol{\delta}_n^s)|^2)$ and space complexity $O(n^2)$.

From Corollary 1, Lemma 4, and Algorithm 3, one way to find all single closed loops for any DFSM is given in Algorithm 4. Algorithm 4 has time complexity $O(|S|)$ and space complexity $O(n^2)$.

Remark 1 From Algorithm 4, it is easy to find that the reason why Definition 2, as the weak version of Definition 1, is applicable in this paper—we traverse

Algorithm 3 Finding θ_α for M

Input: $\boldsymbol{T}$, α, $S:=\{i\,||\Psi(\mathrm{col}_i(\boldsymbol{T}))|>1\}$, $D:=S\backslash\alpha$, $\theta_\alpha:=\{\alpha\}$, $L:=\varnothing$
Output: θ_α

```
while θ_α ∪ L ≠ S do
    T* := T
    Y := Ψ(x)|(T − I_{n×n})x = 0
    for i ∈ D do
        for δ_n^j ∈ Ψ(Tδ_n^i) do
            if j ∉ θ_α then
                T*(:, j) := 0
                T*(α, j) := T*(α, j) + 1
                for δ_n^k ∈ Ψ(Tδ_n^i)\δ_n^j do
                    T*(k, j) := T*(k, j) − 1
                end for
            else
                T*(j, j) := T*(j, j) + 1
            end if
        end for
        T*(α, α) := T*(α, α) + 1
        if (T* − I_{n×n})x = Tδ_n^{θ_α} has a solution set C and
            |{k | k ∈ C ∧ k ∩ Y = ∅}| = |Ψ(Tδ_n^i)| then
            BREAK: θ_α := θ_α ∪ {x_i}
        else
            L := L ∪ {x_i}
        end if
    end for
end while
```

all bifurcation states and detect the existence of a single closed loop at each bifurcation state.

3.2 Static approach for a controllable equivalent form

Clearly, for any DFSM, all states in a single closed loop are mutually controllable. Therefore, we can combine a single closed loop into a single "aggregate" state without changing the controllability of the whole DFSM. We give the following definition:

Definition 4 Consider a DFSM $M = (X = \Delta_n, f_m, x_0)$. $A = (X = \Delta_a, f_a, x_0)$ is called a controllable equivalent form of M, if there exists a function $f : \Delta_a \to 2^{\Delta_n}$, such that

$$f(f_m(x, e)) = f_a(f(x), e).$$

The function f shows the correspondence between the new state $\boldsymbol{y}_i \in \Delta_a$ and the original state $\boldsymbol{x}_i \in \Delta_n$, and is called a controllable equivalent function. We are used to expressing a DFSM in terms of state transition matrices. One way to obtain a controllable equivalent form in the TM model is stated in Algorithm 5. Note that Algorithm 5 can be considered as a proposition for the controllable equivalent form. Therefore, we present a part of the MATLAB code to emphasize this point. Algorithm 5 has time complexity $O(l)$ and space complexity $O(n^2)$, where l is the number of single closed loops in the DFSM. Given a set $S \neq \varnothing$ of which the elements are sets, $\pi(S)$ is defined as $\pi(S) = (S \backslash x_i, x_j) \cup (x_i \cup x_j)$ where $x_i, x_j \in S$ and $x_i \cap x_j \neq \varnothing$. For a DFSM, the controllable equivalent form is often not unique. In practice, we are more concerned with the DFSM with the fewest states. Thus, we give the following definition:

Algorithm 4 Finding all single closed loops for M

Input: $\boldsymbol{T}$, $W := \varnothing$, $H := \varnothing$, $D := \varnothing$, $S := \{i \,|\, |\Psi(\mathrm{col}_i(\boldsymbol{T}))| > 1\}$
Output: W_M
while $S \neq D \cup H$ **do**
  $\alpha \in S \backslash H$
  Apply Algorithm 3 to obtain θ_α
  if $|\theta_\alpha| = 1$ **then**
    $D := D \cup \theta_\alpha$
  else
    $H := H \cup \theta_\alpha$
  end if
  $C := \left\{ \boldsymbol{x} \middle| (\widehat{\boldsymbol{T}} - \boldsymbol{I}_{n\times n})\boldsymbol{x} = \boldsymbol{T}\boldsymbol{\delta}_n^S \right\}$
  for $\boldsymbol{i} \in C$ **do**
    if $(\boldsymbol{T}\boldsymbol{i}).*\boldsymbol{i} = \boldsymbol{i} \wedge (\boldsymbol{T}^{\mathrm{T}}\boldsymbol{i}).*\boldsymbol{i} = \boldsymbol{i}$ **then**
      $W := W \cup \{\Psi(\boldsymbol{i})\}$
    end if
  end for
end while
$C^* := \{\Psi(\boldsymbol{x}) | (\boldsymbol{T} - \boldsymbol{I}_{n\times n})\boldsymbol{x} = \boldsymbol{0} \wedge (\boldsymbol{T} - \boldsymbol{I}_{n\times n})\boldsymbol{x}^* \neq \boldsymbol{0}$ $(x^* \in 2^x)\}$
$W := W \cup C^*$

Definition 5 Consider a DFSM $M = (X = \Delta_n, f_m, x_0)$. $A = (X = \Delta_a, f_a, x_0)$ is one of the controllable equivalent forms of M. A is called a minimal controllable equivalent form for M, if $a = n - \theta + p$ where $\theta = |\bigcup_{x \in W_M} x|$ and $p = |\pi(W_M)|$.

Algorithm 5 Controllable equivalent form of M

Input: $\boldsymbol{T}$, W_M, $\boldsymbol{T}^* := [\ ]$, $\boldsymbol{T}^\circ := [\ ]$
Output: $\boldsymbol{T}^\circ$ is the TM model of the controllable equivalent form of M; the controllable equivalent function f is defined as $y_i = f(x_i) = \Psi(\mathrm{col}_i(\boldsymbol{T}^*))$
for $i \in W_M$ **do**
  $a = \min\left(\mathrm{find}\left(\boldsymbol{\delta}_n^i\right)\right)$
  $\boldsymbol{1}_n = \mathrm{ones}(n,1)$
  $\boldsymbol{b} = \boldsymbol{1}_n - \boldsymbol{\delta}_n^i$
  $\boldsymbol{T}^* = \mathrm{diag}(\boldsymbol{b})$
  $\boldsymbol{T}^*(:, a) = \boldsymbol{\delta}_n^i$
  $\boldsymbol{T}^*(:,\ \mathrm{all}(\boldsymbol{T}^* == 0)) = [\ ]$
  $\boldsymbol{A} = ((\boldsymbol{T}.') * \boldsymbol{T}^*)$
  $\boldsymbol{A}(:, a) = \boldsymbol{A}(:, a) - \boldsymbol{\delta}_n^i$
  $\boldsymbol{T}^\circ = ((\boldsymbol{A}.') * \boldsymbol{T}^*)$
end for

To obtain the minimal controllable equivalent form, compound closed loops must be processed. There are two ways to solve the compound closed loops: the circulation method and the virtual state method. The circulation method detects a single closed loop and combines the states therein repeatedly. Algorithm 6 is proposed with the circulation method and has time complexity $O(c)$ and space complexity $O(n^2)$, where c is the maximum number of single closed loops nested in a compound closed loop.

Remark 2 Note that Algorithm 6 obtains all closed loops in a DFSM while obtaining the minimal controllable equivalent form. Although its

Algorithm 6 Finding all closed loops and the minimal controllable equivalent form for M

Input: $\boldsymbol{T}$, $\boldsymbol{T}_{\min} := \boldsymbol{T}$, $W^* := \varnothing$
Output: $\boldsymbol{T}_{\min}^\circ$ is the TM model of the minimal controllable equivalent form; W^* contains all closed loops in M
repeat
  $\boldsymbol{T}_{\min} := \boldsymbol{T}_{\min}^\circ$
  Apply Algorithm 4 to obtain $W_{\min}$ for $\boldsymbol{T}_{\min}$
  Apply Algorithm 5 to obtain $\boldsymbol{T}_{\min}^\circ$ for $\boldsymbol{T}_{\min}$
  $W^* := W^* \cup W_{\min}$
until $\boldsymbol{T}_{\min}^\circ = \boldsymbol{T}_{\min}$

efficiency decreases with an increase in the number of compound closed loops, it is still valuable when performing distributed simplification of large networks of DFSMs.

3.3 Static approach for controllability

In this subsection, we reconsider the issue of controllability with the help of closed loops.

Lemma 5 Consider DFSM $M = (X = \Delta_n, E, f, x_0)$. If $\boldsymbol{x}_a \in \Delta_n$ is controllable to $\boldsymbol{x}_b \in \Delta_n$, then there exists a closed loop containing x_a and x_b in $\boldsymbol{M}^\kappa$, where $\boldsymbol{M}^\kappa$ is defined as

$$M^\kappa(i,j) = \begin{cases} 1, \text{ if } i = a \wedge j = b, \\ T(i,j), \text{ otherwise.} \end{cases}$$

Proof By contradiction, assume that there is no closed loop. We then have that x_a and x_b are either unreachable to each other or reachable in one direction. Note that $M^\kappa(a,b) = 1$ implies that x_b is controllable to x_a. Hence, x_a is not controllable to x_b, and a contradiction holds.

To cope with the problem of repeated matrix product, we now present a procedure that virtualizes a connection from the goal state to the start state, and uses the idea of closed loops with a virtual state method. This procedure is called the virtual connection method and is reported in Algorithm 7. The so-called virtual state method refers to adding a virtual state to each closed loop, to destroy the structure of the closed loops that are nested in compound closed loops. At this time, the compound closed loops become single closed loops. More precisely, given $M = (X = \Delta_n, E, f, x_0)$ and Q ($Q \subseteq X$), the TM model with the virtual state method for Q, denoted by $\boldsymbol{T}_{V(Q)}$, is defined as

$$\begin{cases} \boldsymbol{T}_{V(Q)}(i,j) = 0 \wedge \boldsymbol{T}_{V(Q)}(l,j) = T_{V(Q)}(i,l) = 1, \\ \qquad \text{if } i \in H^* \wedge x_j \in Q \wedge T(i,j) = 1, \\ \boldsymbol{T}_{V(Q)}(i,j) = 0, \text{ if } i,j > n, \\ \boldsymbol{T}_{V(Q)}(i,j) = T(i,j), \text{ otherwise,} \end{cases}$$

where $H^* = H \cap \bar{S}$, $\boldsymbol{\delta}_n H = \Psi(\boldsymbol{T}\boldsymbol{\delta}_n^Q)$, $\bar{S} = \left\{i \,\middle|\, \left|\Psi(\text{col}_i(\boldsymbol{T}^\text{T}))\right| > 1\right\}$, the dimension of $\boldsymbol{T}_{V(Q)}$ is $(n+|H^*|)\times(n+|H^*|)$, and $l \in \Delta_{V(Q)} = \Delta_{n+|H^*|} \backslash \Delta_n$ is the corresponding virtual state.

Remark 3 If $\boldsymbol{T}_{V(S)}$ and $\widehat{\boldsymbol{T}}_{V(S)}$ are constructed in advance, the time complexity of Algorithm 7 can be reduced to $O(1)$.

Algorithm 7 Two-state controllability with the virtual connection method for $M = (X = \Delta_n, E, f, x_0)$

Input: $\boldsymbol{T}$, goal state x_b, start state x_a, $S := \{i \,||\Psi(\text{col}_i(\boldsymbol{T}))| > 1\}$
$\boldsymbol{T}(x_a, x_b) = 1$
Construct $\boldsymbol{T}_{V(S)}$
Construct $\widehat{\boldsymbol{T}}_{V(S)}$ where α is specified as the start state x_a
$C := \left\{\Psi(\boldsymbol{x}) \,\middle|\, (\widehat{\boldsymbol{T}}_{V(S)} - \boldsymbol{I}_{(n+|H^*|)\times(n+|H^*|)})\boldsymbol{x} = \boldsymbol{T}_{V(S)}\boldsymbol{\delta}_{n+|H^*|}^S \wedge (\boldsymbol{T}\boldsymbol{x}).*\boldsymbol{x} = \boldsymbol{x} \wedge (\boldsymbol{T}^\text{T}\boldsymbol{x}).*\boldsymbol{x} = \boldsymbol{x}\right\}$
if $C \neq \varnothing$ **then**
  BREAK: x_a is controllable to x_b
else
  BREAK: x_a is not controllable to x_b
end if

4 An illustrative example

Example 1 Consider $A = (X, E, f, x_0)$ depicted in Fig. 1, where $X = \{1,2,3,4,5,6,7,8,9,10\}$, $x_0 = \{1\}$, $E = \{a, b\}$, and transition function f is represented by labeled arrows.

1. Apply Algorithm 3 to obtain θ_α where $\alpha = 5$, and apply Algorithm 5 to obtain θ.

The TM model of A is as follows:

$$\boldsymbol{T} = \boldsymbol{\delta}_{10}\left[2, \{3,7\}, \{4,8\}, 5, \{3,6\}, 10, 5, 9, 10, 10\right].$$

Then we have $S = \{2,3,5\}$, $D = \{2,3\}$, $\theta_\alpha = \{5\}$, and $Y = \{10\}$. For $i = 3 \in S$, $\boldsymbol{T}^*$ is constructed as

$$\boldsymbol{T}^* = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

By solving $(\boldsymbol{T}^* - \boldsymbol{I}_{10\times 10})\boldsymbol{x} = \boldsymbol{T}\boldsymbol{\delta}_{10}^{\{5\}}$, we have $x_1 = \{3,4,5\}$, $x_2 = \{3,5,8\}$, $x_3 = \{3,4,5,10\}$, and $x_4 = \{3,5,8,10\}$. Because $|\{k\,|k \in C \wedge k \cap Y = \varnothing\}| = |\{x_1, x_2\}| = 2 = \left|\Psi(\boldsymbol{T}\boldsymbol{\delta}_n^{i=3})\right|$ where $C = \{x_1, x_2, x_3, x_4\}$, we obtain $\theta_\alpha = \{3,5\}$. For $i = 2 \in S$, constructing $\boldsymbol{T}^*$ and solving $(\boldsymbol{T}^* - \boldsymbol{I}_{10\times10})\boldsymbol{x} = \boldsymbol{T}\boldsymbol{\delta}_{10}^{\{3,5\}}$, $x = \{10\}$. Because $|\{k\,|k \in C = \{x\} \wedge k \cap Y = \varnothing\}| = 0 \neq 2 = \left|\Psi(\boldsymbol{T}\boldsymbol{\delta}_n^{i=2})\right|$, we obtain $L = \{2\}$. Note that $\theta \cup L = S$. Thus, the algorithm terminates and $\theta_{\alpha=5} = \{3,5\}$. Moreover, we have $\theta_{\alpha=5} = \theta$ because $H = \theta$, $D = L$, and $D \cup H = S$, which implies that there is only one closed loop in A.

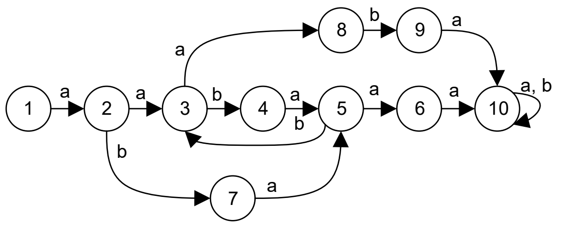

Fig. 1 DFSM A

2. Apply Algorithm 6 to obtain the minimal controllable equivalent form.

According to the above subquestion, we have $\theta = \{3,5\}$. By solving $(\widehat{\boldsymbol{T}} - \boldsymbol{I}_{10\times10})\boldsymbol{x} = \boldsymbol{T}\boldsymbol{\delta}_{10}^{S}$, we have $W_M = \{3,4,5\}$. By applying Algorithm 5, we obtain

$$\boldsymbol{T}^\circ = \boldsymbol{\delta}_8\,[2,\{3,5\},\{4,6\},8,3,7,8,8],$$
$$\boldsymbol{T}^* = \boldsymbol{\delta}_{10}\,[1,2,\{3,4,5\},6,7,8,9,10].$$

According to the controllable equivalent function $f(x_i) = \Psi(\mathrm{col}_i(\boldsymbol{T}^*))$ and the notation $y_i = f(x_i) = x_i'$, we have $1' = \{1\}$, $2' = \{2\}$, $3' = \{3,4,5\}$, $4' = \{6\}$, $5' = \{7\}$, $6' = \{8\}$, $7' = \{9\}$, $8' = \{10\}$. The minimal controllable equivalent form is shown in Fig. 2.

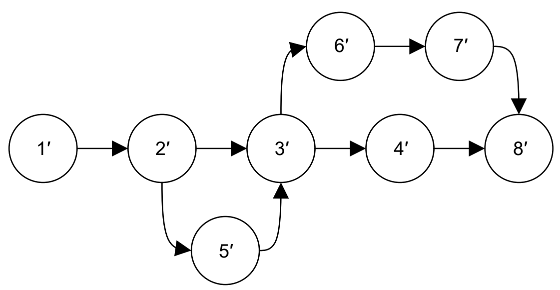

Fig. 2 The minimal controllable equivalent form of A in the TM model

3. Apply Algorithm 7 to check the two-state controllability from state 4 to state 9.

We first make $T(4,9) = 1$ to virtualize a connection from state 9 to state 4. Based on $\boldsymbol{T}$, we have $S = \{2,3,5\}$, $\bar{S} = \{3,4,5,10\}$, $\boldsymbol{\delta}_{10}H = \boldsymbol{\delta}_{10}\,\{3,4,6,7,8\}$, $\boldsymbol{\delta}_{10}H^* = \boldsymbol{\delta}_{10}\,\{3,4\}$. Because $\Psi(\boldsymbol{T}\boldsymbol{\delta}_{10}^3) \cap \boldsymbol{\delta}_{10}H^* = \{\boldsymbol{\delta}_{10}^4\}$ and $\Psi(\boldsymbol{T}\boldsymbol{\delta}_{10}^5) \cap \boldsymbol{\delta}_{10}H^* = \{\boldsymbol{\delta}_{10}^3\}$, virtual states 11, 12, and 13 are proposed such that $2 \to 3 \Rightarrow 2 \to 11 \to 3$, $3 \to 4 \Rightarrow 3 \to 12 \to 4$, $5 \to 3 \Rightarrow 5 \to 13 \to 3$. The modified TM model is shown in Fig. 3, where the dashed lines refer to the virtual states and the virtual connection. It is easy to obtain $\theta_{\alpha=9} = \{3,5,9\}$ by Algorithm 3. By solving $(\widehat{\boldsymbol{T}}_{V(S)} - \boldsymbol{I}_{13\times13})\boldsymbol{x} = \boldsymbol{T}_{V(S)}\boldsymbol{\delta}_{13}^{S=\{2,3,5\}}$, we have $\boldsymbol{x} = [0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1]^{\mathrm{T}} \sim \{3,4,5,8,9,13\}$. Thus, there exists a closed loop between states 4 and 9, which implies that state 4 is controllable to state 9.

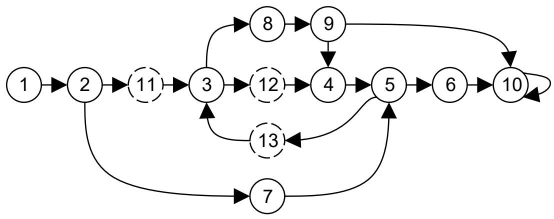

Fig. 3 The modified TM model of A with the virtual states

5 Concluding remarks

A matrix-based static approach for detection of a closed loop has been proposed. Based on the static view, we propose the definitions of the controllable equivalent form and minimal controllable equivalent form. The static approach is then extended for controllability and eliminates the "explosion of complexity" problem inherent in the existing approaches. For the issues mentioned in this work, the implementation of our algorithms is much simpler than that of algorithms designed from the dynamic process perspective.

Contributors

He DENG conceived the concept, designed the research, and drafted the paper. Yongyi YAN and Zengqiang CHEN supervised the research, helped organize the paper, and revised and finalized the paper.

Compliance with ethics guidelines

He DENG, Yongyi YAN, and Zengqiang CHEN declare that they have no conflict of interest.

References

Chen ZQ, Zhou YR, Zhang ZP, et al., 2020. Semi-tensor product of matrices approach to the problem of fault detection for discrete event systems (DESs). *IEEE Trans Circ Syst II Expr Briefs*, 67(12):3098-3102. https://doi.org/10.1109/TCSII.2020.2967062

Cheng DZ, Qi HS, 2010. A linear representation of dynamics of Boolean networks. *IEEE Trans Autom Contr*, 55(10): 2251-2258. https://doi.org/10.1109/TAC.2010.2043294

Han XG, Chen ZQ, Liu ZX, et al., 2018. The detection and stabilisation of limit cycle for deterministic finite automata. *Int J Contr*, 91(4):874-886. https://doi.org/10.1080/00207179.2017.1295319

Lu JQ, Li HT, Liu Y, et al., 2017. Survey on semi-tensor product method with its applications in logical networks and other finite-valued systems. *IET Contr Theory Appl*, 11(13):2040-2047. https://doi.org/10.1049/iet-cta.2016.1659

Lu JQ, Sun LJ, Liu Y, et al., 2018. Stabilization of Boolean control networks under aperiodic sampled-data control. *SIAM J Contr Optim*, 56(6):4385-4404. https://doi.org/10.1137/18M1169308

Xu Q, Zhang ZP, Yan YY, et al., 2021. Security and privacy with K-step opacity for finite automata via a novel algebraic approach. *Trans Inst Meas Contr*, 43(16):3606-3614. https://doi.org/10.1177/01423312211040314

Xu XR, Hong YG, 2013a. Matrix approach to model matching of asynchronous sequential machines. *IEEE Trans Autom Contr*, 58(11):2974-2979. https://doi.org/10.1109/TAC.2013.2259957

Xu XR, Hong YG, 2013b. Observability analysis and observer design for finite automata via matrix approach. *IET Contr Theory Appl*, 7(12):1609-1615. https://doi.org/10.1049/iet-cta.2013.0096

Yan YY, Chen ZQ, Liu ZX, 2014. Semi-tensor product of matrices approach to reachability of finite automata with application to language recognition. *Front Comput Sci*, 8(6):948-957. https://doi.org/10.1007/s11704-014-3425-y

Yan YY, Deng H, Chen ZQ, 2021. A new look at the critical observability of finite state machines from an algebraic viewpoint. *Asian J Contr*, early access. https://doi.org/10.1002/asjc.2705

Yan YY, Yue JM, Chen ZQ, 2022. Observed data-based model construction of finite state machines using exponential representation of LMs. *IEEE Trans Circ Syst II Expr Briefs*, 69(2):434-438. https://doi.org/10.1109/TCSII.2021.3087189

Yue JM, Yan YY, Chen ZQ, 2019. Language acceptability of finite automata based on theory of semi-tensor product of matrices. *Asian J Contr*, 21(6):2634-2643. https://doi.org/10.1002/asjc.2190

Zhu R, Chen ZQ, Zhang JL, et al., 2022. Strategy optimization of weighted networked evolutionary games with switched topologies and threshold. *Knowl-Based Syst*, 235:107644. https://doi.org/10.1016/j.knosys.2021.107644

Zhu SM, Feng JE, Sun LY, 2021. Matrix expression of Owen values. *Asian J Contr*, early access. https://doi.org/10.1002/asjc.2738