SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Stabilization of a class of congestion games via intermittent control

Kaichen Jiang & Jinhuan Wang*

School of Science, Hebei University of Technology, Tianjin 300401*, P. R. China*

Appendix A Semi-tensor product of matrices

Here we briefly introduce the definition and some useful properties of semi-tensor product (STP) of matrices. More details please refer to [1].

Definition 1. Let $X \in \mathcal{M}_{m\times n}, Y \in \mathcal{M}_{p\times q}$. Define the STP of X and Y as

$$X \ltimes Y := (X \otimes I_{\frac{r}{n}})(Y \otimes I_{\frac{r}{p}}),$$

where $\otimes$ represents the Kronecker product, $r = lcm\{n, p\}$ is the least common multiple of n and p.

Proposition 1. Given $Y \in \mathcal{M}_{m\times n}$. Let $X \in \mathbb{R}^t$ be a column vector. Then

$$XY = (I_t \otimes Y)X.$$

Proposition 2. Let $X \in \mathbb{R}^m, Y \in \mathbb{R}^n$ be two column vectors. Then

$$W_{[m,n]}XY = YX,$$

where $W_{[m,n]}$ is the swap matrix defined as

$$W_{[m,n]} = \delta_{mn}[1, m+1, \dots, (n-1)m+1;\ 2, m+2, (n-1)m+2; \dots;\ m, 2m, \dots, nm].$$

Lemma 1. Let $f(x_1, x_2, \dots, x_n) : \mathcal{D}_k^n \to \mathcal{D}_k$ be a k-valued logical function, where $\mathcal{D}_k := \{1, 2, \dots, k\}$ and $\mathcal{D}_k^n := \mathcal{D}_k \times \cdots \times \mathcal{D}_k$. Then there exists a unique matrix $M_f \in \mathcal{L}_{k\times k^n}$ such that

$$f(x_1, x_2, \dots, x_n) = M_f \ltimes_{i=1}^n x_i.$$

Appendix B Proof of Theorem 2

Proof. From the system (7), when the control is not triggered, one has $x(t) = Lx(t-1) = L^2x(t-2) = \cdots = L^t x(0)$. Assume the control is free-input and given as $\{u(t) \in \Delta_{K^l}\}_{t=t_1,t_2,\dots,t_\rho}$, then one has

$$\begin{aligned} x(t_\rho + 1) &= \tilde{L}x(t_\rho)u(t_\rho) = \tilde{L}L^{t_\rho - t_{\rho-1}-1}\tilde{L}x(t_{\rho-1})u(t_{\rho-1})u(t_\rho) \\ &= \cdots = \tilde{L}L^{t_\rho - t_{\rho-1}-1} \cdots \tilde{L}L^{t_2-t_1-1}\tilde{L}L^{t_1}x(0)u(t_1)u(t_2)\dots u(t_\rho). \end{aligned}$$

One sees that for any $x(0) \in \Delta_K$, $x(t_\rho + 1) \in S_{uc}^0$ under the control sequence $\{u(t) \in \Delta_{K^l}\}_{t=t_1,t_2,\dots,t_\rho}$ if and only if $\mathrm{Col}_j(\tilde{L}L^{t_\rho - t_{\rho-1}-1} \cdots \tilde{L}L^{t_2-t_1-1}\tilde{L}L^{t_1}x(0)) \in S_{uc}^0$, for some $j \in \{1, 2, \dots, K(K^l)^\rho\}$. That is for any $x(0) \in \Delta_K$, it will evolve to a state in S_{uc}^0 under the above control sequence, and then it will evolve naturally to the NE $\delta_K^{j^*}$. This completes the proof.

Appendix C Proof of Theorem 3

Before prove the theorem, one needs to review the sets S_c^i and S_{uc}^i firstly,

$$S_c^i = \{\delta_K^j \in \Delta_K \backslash \bigcup_{\mu=0}^{i-1} \tilde{S}^\mu \mid \tilde{S}^{i-1} \cap \mathrm{Col}(\tilde{L}_j) \neq \emptyset\}, \tag{C1}$$

$$S_{uc}^i = \{\delta_K^j \in \Delta_K \backslash ((\bigcup_{\mu=0}^{i-1} \tilde{S}^\mu) \cup S_c^i) \mid \exists\ \tau \in\ [1, K-1]\ \text{s.t.}\ L^\tau \delta_K^j \in S_c^i\}. \tag{C2}$$

* Corresponding author (email: wjhuan228@163.com)

Proof of Theorem 3. (Necessity): Assume the system (7) can be stabilized to $\delta_K^{j^*}$ globally under control $u = Hx(t)$. Assume that for any $T \in [1, K-1], \bigcup_{i=0}^{T} \tilde{S}^i \neq \Delta_K$. From the construction of the sets we know as T gets larger, $\bigcup_{i=0}^{T} \tilde{S}^i$ gets larger. Then from the global stabilization, there exists a $t \in [K, \infty)$ such that $\bigcup_{i=0}^{t} \tilde{S}^i = \Delta_K$. From Figure 1, one sees that for $i \geqslant 1$, any state in $\tilde{S}^i$ can be stabilized to the NE by i steps of the control, and the states under each step of the control are different during the evolutionary process. Then one has $| \bigcup_{i=1}^{t} \tilde{S}^i | \geqslant t \geqslant K$. Notice that the set $\tilde{S}^0 \neq \emptyset$, and the number of profiles in system (8) is K, one has $| \bigcup_{i=0}^{t} \tilde{S}^i | > K$, which is a contradiction.

(Sufficiency): Assume $\bigcup_{i=0}^{T} \tilde{S}^i = \Delta_K$ holds for some $T \in [1, K-1]$. We are going to prove the sufficiency by mathematical induction.

When $i = 0$, it is clearly that for any $\delta_K^j \in \tilde{S}^0$, δ_K^j can evolve naturally to the NE $\delta_K^{j^*}$.

When $i = 1$, for any $\delta_K^j \in S_c^1$, from (C1), there correspondingly exists a q such that $\mathrm{Col}_q \tilde{L}_j \in \tilde{S}^0$, which means the profile δ_K^j can be stabilized to $\tilde{S}^0$ by one step control. For any $\delta_K^j \in S_{uc}^1$, from (C2), it can evolve naturally to S_c^1. Therefore, any profile in $\tilde{S}^1$ can converge to $\tilde{S}^0$, then to the NE finally.

Suppose the conclusion is still true for $i = p,\ p \in [1, T-1]$, that is, any profile in $\tilde{S}^p$ can converge to the NE.

When $i = p+1$, for any $\delta_K^j \in S_c^{p+1}$, from (C1), one can see that the profile δ_K^j can be stabilized to $\tilde{S}^p$ by one step control. And for any $\delta_K^j \in S_{uc}^{p+1}$, the profile δ_K^j can evolve naturally to S_c^{p+1} from (C2). Therefore, any profile in $\tilde{S}^{p+1}$ can converge to $\tilde{S}^p$. Recall that all profiles in $\tilde{S}^p$ can converge to the NE, then any profile in $\tilde{S}^{p+1}$ can converge to the NE finally.

In conclusion, all profiles in $\bigcup_{i=0}^{T} \tilde{S}^i$ can be stabilized to the NE. Notice that $\bigcup_{i=0}^{T} \tilde{S}^i = \Delta_K$, then the dynamic system (7) can be globally stabilized to the NE $\delta_K^{j^*}$. The proof is completed.

Appendix D Design of the gain matrix

Algorithm D1 Design of the gain matrix H

Require: K,T,K^l;
set $H = \delta_{K^l}[v_1, v_2, \ldots, v_K]$;
for $j = 1$ to K;
for $i = 0$ to T;
if $\delta_K^j \in \tilde{S}^i$ **then**
  if $\delta_K^j \in S_{uc}^i$ **then**
    v_j takes values randomly from $\delta_{K^l}\{1, 2, \ldots, K^l\}$;
  else
    $v_j = \max\{v \mid \mathrm{Col}_v(\tilde{L}_j) \subseteq \tilde{S}^{i-1}\}$;
  end if
end if
return H

Appendix E Illustrative example

Network function virtualization (NFV) is a very promising research topic in recent years. It divides the functions of the network nodes into blocks by using virtualization technology, and then implements them in software rather than hardware architecture to create network services. This technique has been widely used in cloud computing, 5G networks and so on. For verifying the practicality of the proposed evolutionary WCGPSU, we consider the servers allocation problem in the NFV. More details on NFV can be referred to [2, 3].

In an NFV, assume three users share four servers in one network service flow. The relationship schema is shown in Figure E1. User A chooses one server to achieve his network transmission, user B chooses two servers and user C chooses three. Obviously, one has $K_1 = 4, K_2 = 6, K_3 = 4$, then $K = 4 \times 6 \times 4 = 96$. Without loss of generality, we arrange the strategies of user i in lexicographic order as shown in Table E1. All users' congestion costs are shown in Table E2. Since the users' strategies are different, the data synchronization rate w of each user is also different, and the values are chosen as $w_A = 1, w_B = 2, w_C = 3$. The user-specific costs are shown in Table E3. In this example, the SUR is still the DP-MBRA. Then one constructs the proper state feedback intermittent control u for guaranteeing the global stabilization of the NE.

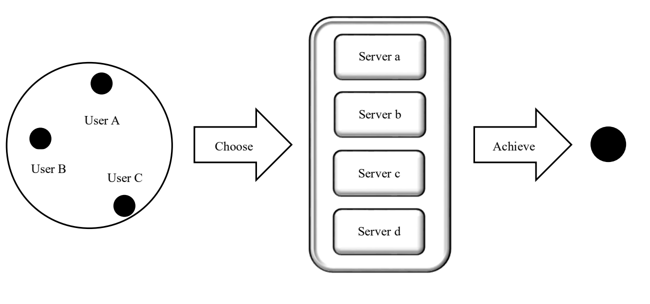

Figure E1 3 users with 4 servers to choose in one flow

Table E1 Strategies of users

Strategy	User A	User B	User C
1	a	a,b	a,b,c
2	b	a,c	a,b,d
3	c	a,d	a,c,d
4	d	b,c	b,c,d
5		b,d	
6		c,d	

Table E2 Congestion cost

Number of users	1	2	3
Server a	3	4	6
Server b	6	8	10
Server c	4	5	8
Server d	2	6	9

Table E3 User-specific cost

User	A	B	C
Server a	3	8	6
Server b	4	6	4
Server c	5	4	4
Server d	6	3	6

From the above discussion, one can convert the dynamic system to algebraic form:

$$x(t+1) = Lx(t), \tag{E1}$$

where $x(t) = \ltimes_{i=1}^{3} x_i(t) \in \Delta_{96}, L = \delta_{96}[95, 23, 23, 23, 96, 24, 24, 24, 21, 21, 21, 21, 22, 22, 22, 22, 23, 23, 23, 23, 21, 21, 21, 21, 95, 23, 23, 7, 95, 23, 23, 7, 23, 23, 23, 7, 23, 23, 23, 7, 23, 23, 23, 7, 23, 23, 23, 7, 83, 23, 11, 11, 82, 22, 10, 10, 9, 21, 9, 9, 10, 22, 10, 10, 11, 23, 11, 11, 10, 22, 10, 10, 95, 7, 7, 7, 93, 5, 5, 5, 21, 5, 5, 5, 23, 7, 7, 7, 21, 5, 5, 5, 21, 5, 5, 5]$.

Then one figures out a fixed point δ_{96}^{21} and a limit cycle $\{\delta_{96}^{5}, \delta_{96}^{96}\}$ of the system. From Theorem 1 one can see that the Nash equilibrium is $s^* = \delta_{96}^{21}$. Actually, due to the decomposition uniqueness of the profile, one can get that $\delta_{96}^{21} = \delta_4^1 \delta_6^6 \delta_4^1$, which means that in the NE, user A chooses Server a, user B chooses Server c,d and user C chooses Server a,b,c.

Adopt the state feedback intermittent control

$$u(t) = Hx(t), t = t_k, \tag{E2}$$

where H is the state feedback gain matrix, t_k is the control instant.

Assume the first player is the controller, then the dynamic system (E1) is converted to:

$$x(t+1) = \begin{cases} L^* u(t) x(t), & x(t) \in \Omega \\ Lx(t), & x(t) \notin \Omega \end{cases} \tag{E3}$$

where $u(t) \in \Delta_4, L^* = I_4 \otimes (L_2 L_3) \in \mathcal{L}_{96 \times 384}$.

By a straightforward calculation, one has

$S_c^0 = S_{uc}^0 = \delta_{96}\{2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 50, 51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 74, 75, 76, 77, 81, 85, 86, 87, 88, 89, 93\}$,

$S_c^1 = \delta_{96}\{1, 5, 25, 29, 49, 53, 73, 78, 79, 80, 82, 83, 84, 90, 91, 92, 94, 95, 96\}$, and $S_{uc}^1 = \emptyset$.

Obviously, one has $\bigcup_{i=0}^{1} \tilde{S}^i = \Delta_{96}$ and $\Omega = S_c^1$. Then from Theorem 3, the system (E3) can be globally stabilized to the NE under the intermittent control (E2). From Algorithm D1 one can figure out one of the state feedback gain matrices as

$$\begin{aligned} H = \delta_4[&3\ 1\ 1\ 1\ 3\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 3\ 1\ 1\ 1\ 3\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ &1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 3\ 1\ 1\ 1\ 3\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ &3\ 1\ 1\ 1\ 1\ 4\ 4\ 4\ 1\ 4\ 4\ 4\ 1\ 1\ 1\ 1\ 1\ 1\ 4\ 4\ 4\ 1\ 4\ 4\ 4]. \end{aligned}$$

To clearly show the evolution of the profile, we randomly choose three different initial strategy profiles, which are simply denoted by $x(0) = (1, 2, 1), x(0) = (4, 6, 4), x(0) = (3, 1, 2)$, respectively. As the profiles evolve, the corresponding total utilities are also changing. The evolutionary process is shown in Figure E2. One sees that all three strategy profiles can converge to the NE $\delta_K^{j*} = \delta_{96}^{21}$ from Figure E2. When the profile converges to the NE, the total costs of all users reach the minimum 87, as shown in Figure E3.

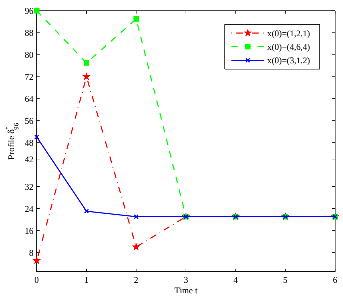

Figure E2 Profile evolution from different initial states.

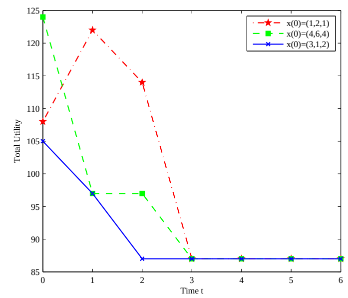

Figure E3 Corresponding total costs

Next, to show the effect of the weight to the game, let $w_A = 3, w_B = 2, w_C = 1$ and other parameters are fixed. Then, one has

$$x(t+1) = Lx(t),$$

where $L = \delta_{96}[95, 71, 71, 23, 96, 72, 24, 24, 69, 69, 69, 21, 95, 23, 23, 23, 21, 21, 21, 21, 21, 21, 21, 21, 95, 71, 71, 7, 95, 71, 23, 7, 69, 69, 69, 5, 95, 23, 23, 7, 23, 23, 23, 7, 21, 21, 21, 5, 83, 71, 59, 11, 82, 70, 10, 10, 57, 69, 57, 9, 82, 22, 10, 10, 9, 21, 9, 9, 9, 21, 9, 9, 93, 53, 53, 5, 93, 53, 5, 5, 69, 53, 53, 5, 93, 5, 5, 5, 21, 5, 5, 5, 21, 5, 5, 5]$.

One can get that there are two fixed points $\delta_{96}^{21}, \delta_{96}^{57}$ and three limit cycles $\{\delta_{96}^{5}, \delta_{96}^{96}\}, \{\delta_{96}^{9}, \delta_{96}^{69}\}, \{\delta_{53}^{5}, \delta_{96}^{83}\}$ of the system. We still adopt the profile δ_{96}^{21} as the target profile and the first player as the controller, then the above dynamic system is converted to (E3).

By a straightforward calculation, one has
$S_c^0 = S_{uc}^0 = \delta_{96}\{7, 15, 17, 21, 22, 23, 24, 28, 31, 32, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 54, 66, 70\}$,
$S_c^1 = \delta_{96}\{1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 16, 18, 19, 20, 25, 26, 27, 29, 30, 33, 34, 35, 37, 50, 62, 89, 93\}$,
$S_{uc}^1 = \delta_{96}\{48, 52, 55, 56, 58, 60, 63, 64, 65, 67, 68, 69, 71, 72, 73, 76, 77, 79, 80, 81, 84, 85, 86, 87, 88, 90, 91, 92, 94, 95, 96\}$,
$S_c^2 = \delta_{96}\{36, 49, 51, 53, 57, 59, 61, 74, 75, 78, 82, 83\}$.

It is easy to get $\bigcup_{i=0}^{2} \tilde{S}^i = \Delta_{96}$ and $\Omega = S_c^1 \cup S_c^2$. Then by Theorem 3, the game can be globally stabilized to the NE δ_{96}^{21} under the intermittent control (E2). The state feedback gain matrix can be chosen as

$$\begin{aligned} H = \delta_4[&1\ 1\ 1\ 1\ 1\ 1\ 4\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 4\ 1\ 4\ 1\ 1\ 1\ 4\ 4\ 4\ 4\ 1\ 1\ 1\ 4\ 1\ 1\ 4\ 4\ 1\ 1\ 1\ 1 \\ &1\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 1\ 1\ 1\ 1\ 4\ 1\ 4\ 4\ 4\ 1\ 4\ 1\ 4\ 1\ 1\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4 \\ &4\ 1\ 1\ 4\ 4\ 1\ 4\ 4\ 4\ 1\ 1\ 4\ 4\ 4\ 4\ 4\ 1\ 4\ 4\ 4\ 1\ 4\ 4\ 4]. \end{aligned}$$

From the above results, one can see that the weight can change the topology structure of the game. By designing different state feedback gain matrix H, the global stabilization of the game can be achieved.

References

1 Cheng D Z, Qi H S, Zhao Y. An introduction to semi-tensor product of matrices and its applications. Singapore, World Scientific, 2012, 23-50
2 Mijumbi R, Serrat J, Gorricho J L, et al. Network function virtualization: state-of-the-art and research challenges. IEEE Commun Surv Tut, 2016, 18: 236-262
3 Hohemberger R, Lorenzon A F, Rossi F, et al. Optimizing distributed network monitoring for NFV service chains. IEEE Commun Lett, 2019, 23: 1332-1336