Demand-side management for a class of smart grid by using STP

Bing Zhu[1] Xiaohua Xia[2] Zhou Wu[3] Yuxuan He[1]

[1]The Seventh Research Division, Beihang University, P.R.China

[2]Department of Electrical, Electronic and Computer Engineering, University of Pretoria, South Africa

[3]School of Automation, Chongqing University, P.R.China

December 2023

Content

Content

Motivation

- Induce rural communities to switch from diesel power to grid power
- Increase energy-efficiency & Protect environment

Motivation

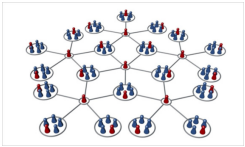
Goal: Persuade users to switch to the grid power

Challenge:

- The price of using grid power is ”dynamic”.
- The initial cost might be high before widely applied.
- No user wants to be the first.

Strategy: A ”networked game” between the supplier and the users.

- Cooperate with a small number of users (controllers), and let users to ”persuade” other users
- Model the process into a control system, where users’ actions are system states
- Networked game based on Semi-Tensor Product (STP)



Contributions

- The demand-side management is modeled into a control networked evolutionary game (NEG) in the framework of semi-tensor product (STP).
- Optimal control policy is introduced to optimize the transient and steady-state performance of the control NEG.
- A state feedback control is proposed to achieve the Nash equilibrium. It is effective in the presence of model uncertainties.
- A robust optimal control problem is formulated, and a solution algorithm is proposed to trade-off robustness and transient performance.

Content

Preliminary

Definition

A *normal finite game* $\mathscr{H}$ can be formulated by three parts:

- the set of players: $\mathscr{V} = \{\pi_1, \pi_2, ..., \pi_p\}$;
- the strategy set for each player: $\mathscr{X}_i = \{x_{i1}, x_{i2}, ..., x_{ik_i}\}$, where $i = 1, 2, ..., p$;
- the cost function for each player: $c_i(x_i, x_{-i})$, where $x_i \in \mathscr{X}_i$ is the strategy selected by player i, and x_{-i} denotes the strategies vector of other players except player i.

Definition

The *networked evolutionary game* is composed by

1. a networked graph $\mathscr{G}$;
2. a normal finite game $\mathscr{H}$ that can be played repeatedly;
3. an updating law Π.

Definition

The *control networked evolutionary game* is composed by

- a networked graph $\mathscr{G}_c = (\mathscr{X} \bigcup \mathscr{U}, \mathscr{E})$, where $\{\mathscr{X}, \mathscr{U}\}$ is a partition of $\mathscr{V}$;
- a normal finite game $\mathscr{H}$;
- an updating law Π.

Example

A typical example of the updating law is Unconditional Imitation:

$$x_i(t+1) = x_{j^*}(t), \;\; j^* = \arg\min_{j \in \mathscr{N}_i} c_j(x_j(t), x_{-j}(t)).$$

Definition

Nash equilibrium (*NE*), denoted by $(x_1^*, x_2^*, ..., x_p^*)$, is a local optimal response for a normal finite game, where no individual would gain by unilaterally changing its own strategy: $c_i(x_i^*, x_{-i}^*) \leq c_i(x_i, x_{-i}^*)$.

Definition

The semi-tensor product of two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$ can be defined by

$$A \ltimes B \triangleq (A \otimes I_{o/n})(B \otimes I_{o/p}) \in \mathbb{R}^{(mo/n)\times(qo/p)}, \tag{1}$$

where $o = \mathrm{lcm}(n,p)$ denotes the least common multiple of n and p; and $\otimes$ denotes the Kronecker product.

Definition

The fundamental vector $\delta_n^i \in \Delta_n = \{\delta_n^1, \cdots, \delta_n^n\}$ is defined as the ith column of the identity matrix $I_{n\times n}$. It can be further defined that

$$\delta_n[i,j,\ldots,k] \triangleq [\delta_n^i, \delta_n^j, \ldots, \delta_n^k] \tag{2}$$

for more compact expressions.

Lemma

With equivalence $i \sim \delta_n^i,\ i = 1,2,\ldots,n$, a logic function $f : \Delta_n^k \to \Delta_n$ can be rewritten by $f(x_1,x_2,\ldots,x_k) = M_f \ltimes_{i=1}^{k} x_i$, where M_f is the structure matrix of logic function f.

Lemma

For a logic dynamic system

$$x_i(t+1) = f_i(x_i(t), x_{-i}(t)) = M_{fi} \ltimes_{i=1}^{n} x_i,\ i = 1, \ldots, n,$$

it can be rewritten in the form of

$$x(t+1) = M_f x(t), \tag{3}$$

where $x(t) \triangleq \ltimes_{i=1}^{n} x_i$, *and*

$$M_f \triangleq M_{f1} * M_{f2} * \cdots * M_{fn}. \tag{4}$$

Here, $*$ *denotes the Khatri-Rao product:*

$$M * N \triangleq [\mathrm{col}_1(M) \ltimes \mathrm{col}_1(N), \cdots, \mathrm{col}_s(M) \ltimes \mathrm{col}_s(N)],$$

where $M \in \mathbb{R}^{p \times s}$ *and* $N \in \mathbb{R}^{q \times s}$; *and* $\mathrm{col}_i(M)$ *denotes the* i*th column of matrix* M. *Moreover, the controlled logic dynamic system*

$$x_i(t+1) = f_i(x_i(t), x_{-i}(t), u(t))\ \ i = 1, \ldots, n,$$

can be rewritten in the form of

$$x(t+1) = Lu(t)x(t). \tag{5}$$

Lemma

For a logic dynamic system

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

δ_n^i *is its fixed point, if and only if the diagonal element* m_{ii} *of* M_f *equals* 1.

Lemma

For logic variable $x \in \mathcal{D}_k$, *it satisfies*

$$x^2 = x \ltimes x = \Phi_N x, \tag{6}$$

where

$$\Phi_N = \ltimes_{j=1}^{k} \left(I_{2^{j-1}} \otimes \left(I_2 \otimes W_{[2,2^{k-j}]} M_r \right) \right), \tag{7}$$

and $W_{[i,j]}$ *is the swap matrix, and* $k = 2^N$.

Problem statement

Consider a smart grid with N users and the network denoted by adjacent matrix $\mathscr{A}$.

- Let $p_i(t)$ denote the price paid by user i at time t.
- $x_i(t)$ is defined as the state (strategy) of user i at time t.
- Each user has the option of using either the grid power ($x_i(t) = \delta_2^1$) or the local diesel power ($x_i(t) = \delta_2^2$).
- The price of the diesel power is fixed at p_d, and the price of the grid power varies according to the number of grid users:

$$p_g(t) = p_g(N_g(t)), \tag{8}$$

  where $N_g(t)$ denotes the number of grid users at time t.
- The price paid by user i is given by

$$p_i(t) = \begin{cases} p_g(t), & \text{if } x_i(t) = \delta_2^1, \\ p_d, & \text{if } x_i(t) = \delta_2^2. \end{cases} \tag{9}$$

- The cost function of each user is defined by

$$c_i(x_i(t), x_{-i}(t)) = p_i(t) + \alpha \left(p_i(t) - \min_{j \in \mathcal{N}_i} p_j(t) \right), \tag{10}$$

where $\alpha > 0$ is a constant weight coefficient.

  * Each user pursues the lowest price.
  * Each user feels uncomfortable if it pays a higher price than those of its neighbors.

- Each user would change its strategy to that of the neighbor with the lowest cost (Unconditional Imitation)

$$x_i(t+1) = x_{j^*}(t), \;\; j^* = \arg\min_{j \in \mathcal{N}_i} c_j(x_j(t), x_{-j}(t)). \tag{11}$$

- The total cost at time t is defined by

$$C(t) = \sum_{i=1}^{N} p_i(t), \tag{12}$$

which is the sum of prices paid by all users.

- It is supposed that some of the users cooperate with the grid provider, such that other users can be induced to switch to grid power.
- The cooperative users are denoted as “controllers”:

$$u_i(t) = x_i(t), \quad \text{if } i \in \mathscr{U}, \tag{13}$$

  and their selection (of using grid or diesel power) can be actively assigned.
- The controllers can be rewritten in a compact form:

$$u = \ltimes_{i \in \mathscr{U}} u_i, \tag{14}$$

  if there are more than one controller.

Objective

1. One *objective* is to design a control algorithm, such that there exist an optimal NE for the networked grid, and the optimal NE can be reached and maintained.

2. It is possible that some users may not report their selections honestly, and other users would be misled. This situation is defined as “uncertainty”. Another *objective* is to find a robust control u, such that the networked grid will still reach and maintain the optimal NE.

Content

Content

Model transformation

Define the state of the network:

$$x = \ltimes_{i=1}^{N} x_i, \tag{15}$$

where $x_i \in \mathcal{D}_2$ denotes the choice of user i.

Based on the cost function $c_i(x)$, a comparison matrix can be defined by

$$C(x) \triangleq \left[c_{ij}(x)\right]_{N\times N}, \tag{16}$$

where the comparison functions are defined by

$$c_{ij}(x) = \begin{cases} \delta_2^1, & \text{if } c_i(x) < c_j(x), \text{ or } j \notin \mathcal{N}_i, \\ \delta_2^2, & \text{if } c_i(x) \geq c_j(x). \end{cases} \tag{17}$$

The comparison functions $c_{ij}(x)$ are logic functions, and their dynamic forms $c_{ij}(x) = C_{ij}x$ can be calculated by using diagrams of true values.

The decision of user i with respect to one of its neighbors j can be described by

$$x_i(t+1) = L_{dij}(c_{ij})x_i(t)x_j(t), \tag{18}$$

where

$$L_{dij}(c_{ij}) = \begin{cases} \delta_2[1,1,2,2], & \text{if } c_{ij} = \delta_2^1, \\ \delta_2[1,2,1,2], & \text{if } c_{ij} = \delta_2^2, \end{cases} \tag{19}$$

or equivalently

$$\begin{aligned} L_{dij}(c_{ij}) =& \delta_2[1,1,2,2,1,2,1,2]c_{ij}(x) = L_d c_{ij}(x) \\ =& L_d C_{ij} x. \end{aligned}$$

The physical implication of (18) is that user i would adopt the strategy of its neighbor j if its cost is larger than or equal to that of j; otherwise, it keeps its strategy.

Define $x_i^+ = x_i(k+1)$ for short. The decision of user i with respect to all its neighbors can be calculated by

$$x_i^+ = L_{diN} \cdots L_{dij} \cdots L_{di1} x_i x_1 \cdots x_j \cdots x_N, \tag{20}$$

where $j \neq i$. Then, with the swap matrix, it holds that

$$x_i^+ = \left(\ltimes_{j=0, j\neq N-i}^{N-1} L_{di,N-j} \right) \left(\ltimes_{j=1, j\neq i}^{i-1} I_{2^{j-1}} \otimes W_{[2,2]} \right) x, \tag{21}$$

where

$$\begin{aligned}
\ltimes_{j=0, j\neq N-i}^{N-1} L_{di,N-j} =& L_d C_{iN} x L_d C_{iN-1} x \cdots L_d C_{i1} x \\
=& L_d C_{iN} \left(I_{2^N} \otimes L_d C_{iN-1} \right) x^2 L_d C_{iN-2} x \cdots L_d C_{i1} x \\
=& L_d C_{iN} \left(I_{2^N} \otimes L_d C_{iN-1} \Phi_N \right) x L_d C_{iN-2} x \cdots L_d C_{i1} x \\
=& L_d C_{iN} \ltimes_{j=1, j\neq N-i}^{N-1} \left(I_{2^N} \otimes L_d C_{iN-j} \Phi_N \right) x.
\end{aligned}$$

Linear dynamics in STP framework

It then follows that

$$x_i^+ = M_i x, \tag{22}$$

where $M_i = M_{i1} M_{i2} \Phi_N$, and

$$M_{i1} \triangleq L_d C_{iN} \ltimes_{j=1, j\neq N-i}^{N-1} \left(I_{2^N} \otimes L_d C_{iN-j} \Phi_N \right), \tag{23}$$

$$M_{i2} \triangleq I_{2^N} \otimes \left(\ltimes_{j=1, j\neq i}^{i-1} I_{2^{j-1}} \otimes W_{[2,2]} \right). \tag{24}$$

It then follows that

$$x^+ = Mx, \tag{25}$$

where $M = M_1 * M_2 * \cdots * M_N$.

Controlled linear dynamics in STP framework

Suppose that r of the N users are controllers:

$$x_{i_1} = u_1, \cdots, x_{i_r} = u_r, \tag{26}$$

and

$$x \triangleq \ltimes_{j=1, x_j \neq u_k}^{N-r} x_j, \quad u \triangleq \ltimes_{j=1}^{r} u_j. \tag{27}$$

It follows from (22) that

$$x_i^+ = M_i \left(\ltimes_{j=1}^{r} W_{[2,2^{i_j - j}]} \right) ux, \quad i \neq i_1, \cdots, i_r, \tag{28}$$

and

$$x^+ = Lux, \tag{29}$$

where

$$L = M_1 \left(\ltimes_{j=1}^{r} W_{[2,2^{i_j - j}]} \right) * \cdots * M_N \left(\ltimes_{j=1}^{r} W_{[2,2^{i_j - j}]} \right). \tag{30}$$

Examples

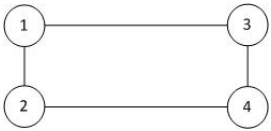


Figure: Communication graph of the communities

Table: Prices of diesel power and grid power

user number	0	1	2	3	4
grid power price	8	7	7	6.5	7.5
diesel power price	7.2	7.2	7.2	7.2	7.2

Note: values in this table are not absolute prices; they are assigned to reflect differences of prices in various scenarios.

Example

For the linear dynamic game, the structural matrix is calculated by

$$M = \delta_{16}[1,1,1,1,1,1,1,2,1,1,1,1,1,1,9,16]. \tag{31}$$

There are two NEs δ_{16}^{1} and δ_{16}^{16}, where all users have the same choice of using grid power or diesel power. However, the cost function is not optimal at both NEs.

Example

Suppose that user 4 is cooperative. The structure matrix can be calculated by

$$L = \delta_{8}[1,1,1,1,1,1,1,5,1,1,1,1,1,2,3,8], \tag{32}$$

and the dynamics of the smart grid is given by $x^{+} = Lux$. There is an optimal NE where $u = \delta_{2}^{1}$, and $x = \delta_{8}^{8}$.

Content

Open-loop optimal policy

An optimal policy can be proposed to

- reach the optimal NE;
- minimize the overall cost in transient process.

The cost function for optimization can be designed by $J \triangleq \sum_{t=0}^{T} C(x(t), u(t))$. The optimization can be formulated by

$$U^* = \arg\min_{U} J \tag{33}$$

s.t.

$$x(t+1) = Lu(t)x(t), \tag{34}$$
$$u(t) = \ltimes u_i(t), \tag{35}$$
$$u_i(t) \in [\delta_2^1, \delta_2^2], \tag{36}$$
$$x(0) = x_0, \tag{37}$$
$$x(T) = x_{NE^*}, \quad \text{(can be removed)} \tag{38}$$

where the optimal solution $U^* = [u^*(0), u^*(1), \ldots, u^*(T)]^T$ is the optimal control policy.

Theorem

Suppose that the following conditions are satisfied:

- *the optimal NE x_{NE^*} is a global optimal point;*
- *with certain control series $[\hat{u}(0), \hat{u}(1), \cdots, \hat{u}(N)]$, the optimal NE x_{NE^*} can be reached within finite time $t = N < T$ from initial states $x(0) = x_0$.*

Then, with sufficiently large control horizon T, the optimal control policy is capable of reaching the optimal NE.

Proof. The result can be proved by contradiction. Assume that, with the control horizon T, the optimal control fails to reach x_{NE^*}.

$$J^* = \sum_{t=0}^{T} C(x^*(t), u^*(t)) = \sum_{t=0}^{N-1} C(x^*(t), u^*(t)) + \sum_{t=N}^{T} C(x^*(t), u^*(t)). \tag{39}$$

where $x^*(t)$ is the corresponding optimal states under optimal control $u^*(t)$; and $x^*(t) \neq x_{NE^*}$.

Proof.(cont'd)
There exists at least another control series $[\hat{u}(0), \hat{u}(1), \cdots, \hat{u}(N), \cdots, \hat{u}(T)]$ such that x_{NE^*} is reached and maintained. The corresponding value of cost function is given by

$$\begin{aligned}\hat{J} &= \sum_{t=0}^{T} C(\hat{x}(t), \hat{u}(t)) = \sum_{t=0}^{N-1} C(\hat{x}(t), \hat{u}(t)) + \sum_{t=N}^{T} C(\hat{x}(t), \hat{u}(t)) \\ &= \sum_{t=0}^{N-1} C(\hat{x}(t), \hat{u}(t)) + (T-N) C(x_{NE^*}, u_{NE^*}).\end{aligned} \tag{40}$$

where u_{NE^*} is the corresponding control to maintain the optimal NE. It then follows that

$$\hat{J} - J^* = \left(\sum_{t=0}^{N-1} C(\hat{x}(t), \hat{u}(t)) - \sum_{t=0}^{N-1} C(x^*(t), u^*(t)) \right) + \sum_{t=N}^{T} \left(C(x_{NE^*}, u_{NE^*}) - C(x^*(t), u^*(t)) \right),$$

where $\sum_{t=0}^{N-1} C(\hat{x}(t), \hat{u}(t)) - \sum_{t=0}^{N-1} C(x^*(t), u^*(t))$ is finite; and $\sum_{t=N}^{T} \left(C(x_{NE^*}, u_{NE^*}) - C(x^*(t), u^*(t)) \right)$ is negative and decreases strictly as T increases.

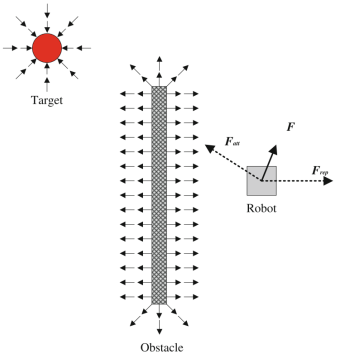
Consequently, $\hat{J} - J^* < 0$ with sufficiently large T, indicating that $\hat{u}(t)$ is superior over $u^*(t)$, which contradicts the assumption given at the beginning of this proof. □

Implementation via Binary Neighborhood Field Optimization (BNFO)

Data: decision variable x_i; objective variable z_i; population size N; problem dimension D; maximum generation G_m; learning rate α; crossover probability Cr;
Result: the best solution in the final population

1 **for** $i \leftarrow 1$ **to** N **do** // Initialization
2 Randomize the ith individual x_i;
3 Repair x_i if it violates the problem's constraints;
4 Calculate x_i's objective value z_i;
5 **end**
6 $G = 1$; // main iteration
7 **while** $G \leq G_m$ **do**
8 **for** $i \leftarrow 1$ **to** N **do**
9 For each individual $x_{i,G}$, find the superior neighbor $xc_{i,G}$ and inferior neighbor $xw_{i,G}$ with respect of Hamming distance; // Location
10 $r_1(\alpha) = rand_1 < \alpha$;
11 $r_2(\alpha) = rand_2 < \alpha$;
12 $v_1 = r_1(\alpha)$&XOR$(xc_{i,G}, x_{i,G})$;
13 $v_2 = r_2(\alpha)$&XOR$(xc_{i,G}, xw_{i,G})$;
14 $v_{i,G} =$ XOR$(x_{i,G}, v_1|v_2)$; // Mutation
15 **for** $j \leftarrow 1$ **to** D **do** // Crossover
16 $u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } rand(0,1) \leq Cr \text{ or } j = j_r \\ x_{j,i,G}, & \text{otherwise} \end{cases}$
17 **end**
18 Repair $u_{i,G}$ if it violates the problem's constraints;
19 $x_{i,G+1} = \begin{cases} u_{i,G}, & \text{if } f(u_{i,G}) \leq f(x_{i,G}) \\ x_{i,G}, & \text{otherwise} \end{cases}$; // Selection
20 **end**
21 $G = G + 1$;
22 **end**

Algorithm 1: Pseudo-code of BNFO



- Initialization
- Location
- Mutation
- Crossover
- Selection

Z. Wu, and T. Chow (2013). Binary neighbourhood field optimisation for unit commitment problems. IET Generation, Transmission& Distribution, 7(3): 298–308.

Z. Wu, and T. Chow (2013). Neighborhood field for cooperative optimization. Soft Computing, 17(5): 819–834.

Content

State Feedback Control via structure shaping

The structure matrix of $x^+ = Lux$ can be shaped by "linear" feedback control

$$u = Gx$$

where the logic matrix $G \in \mathscr{L}_{2^r \times 2^{N-r}}$ is the control gain; and r denotes the number of controllers.

In detail, substituting the feedback control into the dynamics yields

$$x^+ = LGx^2 = LG\Phi_{N-r}x \tag{41}$$

The aim is to design G, such that

$$LG\Phi_{N-r} = M_d, \tag{42}$$

where $M_d = \delta_{2^{N-r}}[m_1, \cdots, m_{2^{N-r}}] \in \mathscr{L}_{2^{N-r} \times 2^{N-r}}$ is the desired structural matrix.

Denote $G = \delta_{2^r}[g_1, g_2, \cdots, g_{2^{N-r}}]$, where $g_i \in [1, 2, 3, \ldots, 2^r]$. It can be calculated directly that

$$G\Phi_{N-r} = (G \otimes I_{2^{N-r}})\Phi_{N-r} = [\delta_{2^r}^{g_1} \otimes I_{2^{N-r}}, \cdots, \delta_{2^r}^{g_{2^{N-r}}} \otimes I_{2^{N-r}}]\Phi_{N-r}, \tag{43}$$

where Φ_{N-r} can be calculated by

$$\Phi_{N-r} = \delta_{2^{2(N-r)}}[1, 1+\Delta, 1+2\Delta, \cdots, 2^{2(N-r)}], \tag{44}$$

with $\Delta = \frac{2^{2(N-r)}-1}{2^{N-r}-1}$, and $2^{2(N-r)} = 1 + (2^{N-r}-1)\Delta$.

The matrix $G\Phi_{N-r}$ is actually the 1st, $(1+\Delta)$th, $(1+2\Delta)$th, $\cdots$ columns of $G \otimes I_{2^{N-r}}$, and it can be written by

$$G\Phi_{N-r} = \delta_{2^N}\left[(g_1 - 1)2^{N-r} + 1, \cdots, (g_{2^{N-r}} - 1)2^{N-r} + 2^{N-r}\right]. \tag{45}$$

Denote $L = \delta_{2^{N-r}}[l_1, l_2, \cdots, l_{2^N}]$, where $l_i \in [1, \cdots, 2^{N-r}]$. It can be re-arranged into a stacked form:

$$\mathbf{L} = \delta_{2^{N-r}} \begin{bmatrix} l_1 & l_2 & l_3 & \cdots & l_{2^{N-r}} \\ l_{2^{N-r}+1} & l_{2^{N-r}+2} & & & l_{2\cdot 2^{N-r}} \\ l_{2\cdot 2^{N-r}+1} & & & & l_{3\cdot 2^{N-r}} \\ \vdots & & & \ddots & \vdots \\ l_{(2^r-1)2^{N-r}+1} & \cdots & & & l_{2^N} \end{bmatrix}. \tag{46}$$

- Each $\delta_{2^{N-r}}^{l_j}$ is the "block element" of the stacked matrix $\mathbf{L}$.
- The ith column of $LG\Phi_{N-r}$ is the g_ith element of the ith column of $\mathbf{L}$.
- The structural matrix $LG\Phi_{N-r}$ of the closed-loop system can be shaped by assigning proper g_i.

Theorem

The structural matrix of logic dynamic system $x^+ = Lux$ can be shaped by feedback control $u = Gx$ into a desired structural matrix, if the columns of the desired structural matrix appear in the corresponding columns of the stacked matrix $\mathbf{L}$.

Example

Consider the structure matrix $L = \delta_8[1,1,1,1,1,1,1,5,1,1,1,1,1,2,3,8]$ of the previous example. Its stacked form is

$$\mathbf{L} = \delta_8 \begin{bmatrix} 1,1,1,1,1,1,1,5 \\ 1,1,1,1,1,2,3,8 \end{bmatrix}. \tag{47}$$

The feedback gain $G = \delta_2[1,2,1,2,1,1,2,2]$ shapes the structural matrix into

$$LG\Phi_3 = \delta_8[1,1,1,1,1,1,3,8]. \tag{48}$$

Example

The previous example can be shaped into the system with only one stable NE. The desired structure matrix can be designed by

$$M_d = \delta_8[1,1,1,1,1,2,3,5], \tag{49}$$

and the control gain can be correspondingly designed by

$$G = \delta_2[1,1,1,1,1,2,2,1]. \tag{50}$$

Algorithm to calculate the control gain G

The design procedure of G can be summarized as follows:

1. Rewrite the structural matrix L into its stacked form **L**;
2. Find the state (the block element) with the lowest cost in **L** (if such block element exists);
3. Find the possible transient states in **L** to satisfy the performance specifications, such that the state with the lowest cost can be reached and maintained.
4. Design G to pick out the desired block elements in steps 2 and 3, such that the desired closed-loop structural matrix M_d can be achieved, and the closed-loop system is capable of reaching and maintaining the NE where the total cost is minimized.

Robustness

Theorem

Consider the controlled logic system $x^+ = Lux$ with uncertainties in its structural matrix L. The uncertainties can be decoupled by the feedback control $u = Gx$, if there exists at least one block element that is not uncertain in each column of the stacked matrix $\mathbf{L}$.

Example

In the previous example, suppose that user 1 reports fake information of its choice to its neighbors. The fake information is $\hat{x}_1 = \delta_2[2,1]x_1$. Then the structural matrix can be calculated by

$$L = \delta_8[*,1,1,1,1,1,1,5,1,*,*,*,1,*,*,8], \quad \mathbf{L} = \delta_8 \begin{bmatrix} *,1,1,1,1,1,1,5 \\ 1,*,*,*,1,*,*,8 \end{bmatrix}. \tag{51}$$

There exists at least one determined element in each column. The control gain can be designed by

$$G = \delta_2[2,1,1,1,2,1,1,1], \tag{52}$$

such that the closed-loop structural matrix is $LG\Phi_3 = \delta_8[1,1,1,1,1,1,1,5]$, and δ_8^1 is still an NE.

Content

Lemma

The finite horizon optimal problem based on cost function

$$J_T(x_0, u(\cdot)) = c_f^{\mathrm{T}} x(T) + \sum_{\tau=0}^{T-1} c^{\mathrm{T}} \ltimes u(\tau) \ltimes x(\tau), \tag{53}$$

where

$$x \in \Delta_{2^n} = \Delta_N, \quad u \in \Delta_{2^m} = \Delta_M, \quad c^{\mathrm{T}} := [\ c_1^{\mathrm{T}} \ | \ c_2^{\mathrm{T}} \ | \ \dots \ | \ c_M^{\mathrm{T}} \ ] \in R^{NM} \tag{54}$$

can be formulated by

$$u^*(\cdot) = \arg\min_{u(\cdot)} J_T(x_0, u(\cdot)), \tag{55}$$

and solved by the following algorithm:

- [*Initialization*] *Set* $\mathbf{m}(T) := c_f$*;*
- [*Recursion*] *For* $t = T-1, ..., 1, 0$*, the* j*th entry of the vector* $\mathbf{m}(t)$ *is chosen using the recursive rule:*

$$[\mathbf{m}(t)]_j := \min_{i \in [1,M]} ([c_i]_j + [\mathbf{m}(t+1)^{\mathrm{T}} L_i])_j, \quad \forall j \in [1, N]. \tag{56}$$

Lemma (Cont'd)

then

- $J_T^*(x_0) := \min_{u(\cdot)} J_T(x_0, u(\cdot)) = \mathbf{m}(0)^{\mathsf{T}} x(0)$.
- *The optimal control input can be implemented by means of a time-varying state feedback law, namely*

$$u(t) = K(t)x(t)$$

where the state feedback matrix is

$$K(t) = [\delta_M^{i^*(1,t)} \quad \delta_M^{i^*(2,t)} \quad \dots \quad \delta_M^{i^*(N,t)}]$$

and

$$i^*(j,t) = \arg \min_{i \in [1,M]} ([c_i]_j + [\mathbf{m}(t+1)^{\mathsf{T}} L_i])_j.$$

E. Fornasini and M. E. Valcher, "Optimal Control of Boolean Control Networks," IEEE Transactions on Automatic Control, vol. 59, no. 5, pp. 1258–1270, May 2014, doi: 10.1109/TAC.2013.2294821.

Theorem

Consider the optimization

$$J(t) \triangleq \sum_{\tau=0}^{T-1} C(u(t+\tau), x(t+\tau)), \quad u^*(\cdot) = \arg\min_{u(\cdot)} J(t) \tag{57}$$

for BCN in case of uncertainties. If the (uncertain) stacked structure matrix has at least one determined element in each of its column, then the optimization is feasible.

Proof. The object function in (57) can be converted into the form:

$$J_T(x_0, u(\cdot)) = c_f^{\mathsf{T}} x(T) + \sum_{\tau=0}^{T-1} c^{\mathsf{T}} \ltimes u(\tau) \ltimes x(\tau)$$

Two issues that need to be considered:

1. How to treat uncertainty
2. Since L appeared in the recursion in the solution process in Fornasini and Valcher's Lemma, do we have to modify this step?

Proof.(cont'd)

1. *How to incorporate the uncertainty into the solution framework of Fornasini and Valcher's Lemma?*

- In each column of **L**, suppose that the row index of the block elements that is not affected by uncertainty can be denoted by

$$i_1^j, i_2^j ..., i_{k_j}^j \in 1, 2, ..., M, \quad k_j \leq M. \tag{58}$$

Using the structure reshaping, it can be seen that for each selectable M_d in **L**, there exists a feedback gain matrix G. These gain matrices can form a set:

$$\Sigma = \{G : Col_j(G) = \delta_M^{g_j}, \quad g_j \in i_1^j, ..., i_{k_j}^j, j \in 1, ..., N\}. \tag{59}$$

- Denote the state of the BCN at time t by $x(t) = \delta_N^s, \ \ s \in \{1, 2, ..., N\}$, and introduce feedback control with gain from Σ: The control u is :

$$\Lambda_s = \{u = G\delta_N^s : G \in \Sigma\}. \tag{60}$$

Proof.(cont'd)

- There is a certain correlation between c in the object function and C

$$[c_i]_j = C(\delta_M^i, \delta_N^j). \tag{61}$$

To avoid a certain transition, it needs to set some entries of c to $+\infty$. Thus, we can modify c by:

$$[c_{ind(u)}]_s = +\infty, \quad u \in \mathscr{L}_M - \Lambda_s,\ s \in \{1, ..N\}, \tag{62}$$

where $ind(u)$ denotes the row index which the element of is equal to 1 in u.

Proof.(cont'd)

2. *Since L appeared in the key step for the recursion in the solution process in Lemma 4.1, do we need to modify this step?*

- The key operation in recursion is:

$$[\mathbf{m}(t)]_j := \min_{i\in[1,M]}([c_i]_j + [\mathbf{m}(t+1)^{\mathrm{T}} L_i])_j, \quad \forall j \in [1,N].$$

It follows that

$$([c_i]_j + [\mathbf{m}(t+1)^{\mathrm{T}} L_i])_j = [c_i]_j + [\mathbf{m}(t+1)^{\mathrm{T}} Col(L_i)_j], \tag{63}$$

- It can be proved that whenever $Col(L_i)_j$ is uncertain, there is $[c_i]_j = +\infty$.
- Consequently, the optimization is not affected by the some uncertain value of $Col(L_i)_j$ and the original L can still be used.

Algorithm

For the robust optimal control problem in the previous theorem, suppose that the row index of the block elements with uncertainty in each column of **L** is

$$\bar{i}^j_1, \bar{i}^j_2 ..., \bar{i}^j_{\hat{k}_j} \in 1, 2, ..., M, \quad \hat{k}_j < M, \quad j \in \{1, 2, ..., N\}, \tag{64}$$

- [Modification] Group elements of the same column into a set Ψ_j, and then replace corresponding items of the vector c in the original optimization problem:

$$[c_r]_j = +\infty, r \in \Psi_j, \quad j \in \{1, ..N\}; \tag{65}$$

- [Generalized recursion] Use Fornasini and Valcher's Lemma 4.1 to solve the robust optimal problem with the modified c and the original L.

Content

Simulation

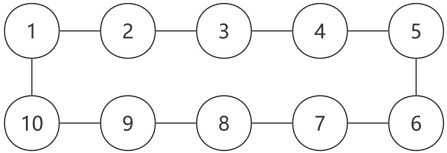


Figure: Topological structure of the smart grid with ten users

- Price of local diesel power: $p_d = 7$.
- Price of grid power:

$$p_g(N_g) = \frac{(N_g - 4.8)^2}{25} + 6.6,$$

where N_g is the number of grid users.

- Updating rule: unconditional imitation

$$x_i(t+1) = x_{j^*}(t), \quad j^* = \arg\min_{j \in \mathcal{N}_i} c_j(x_j(t), x_{-j}(t)).$$

- Users 7 and 10 are controllers u_1 and u_2.

Open-loop optimal policy via BNFO

- Open loop optimal policy with initial state $\mathbf{x}(0) = \delta_{256}^{200}$.

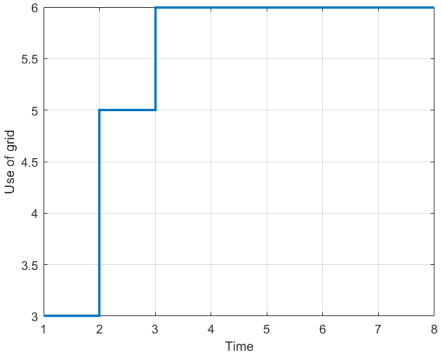


Figure: Number of users using the grid power in E.g. 1

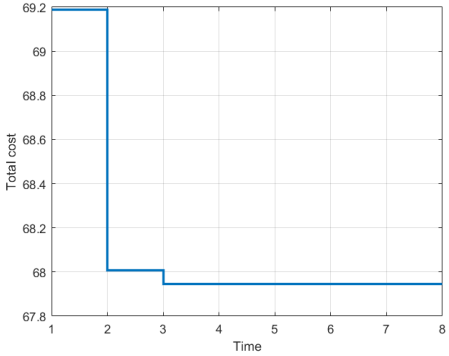


Figure: The overall cost in E.g. 1

State Feedback Control

- State feedback control with initial state: $\mathbf{x}(0) = \delta_{256}^{240}$.

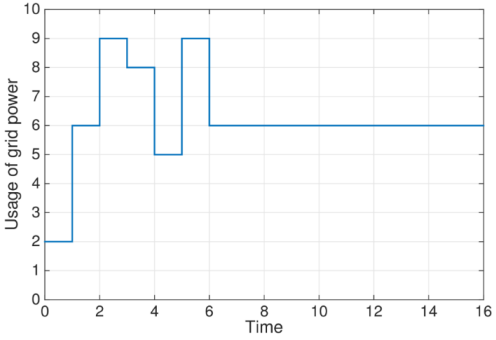


Figure: Number of users using the grid power in E.g. 2

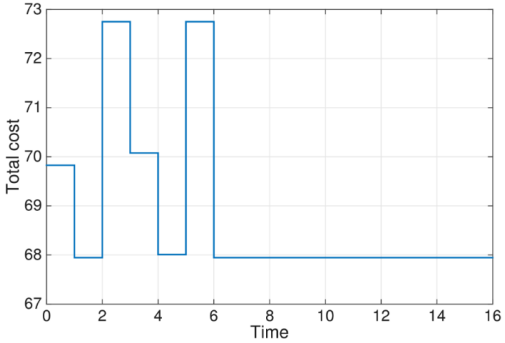


Figure: The overall cost in E.g. 2

Robust Optimal Control

- Robust Optimal control with initial state: $\mathbf{x}(0) = \delta_{256}^{240}$.

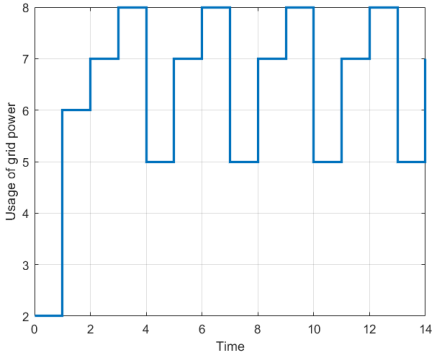


Figure: Number of users using the grid power in E.g. 3

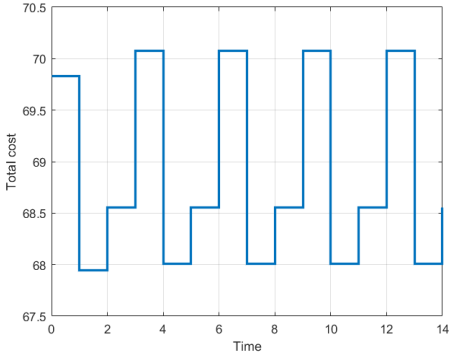


Figure: The overall cost in E.g. 3

Content

Concluding remarks

What we have done?

- Demand-side management modeling in STP framework
- Open loop optimization; state feedback control; robust optimal control

What we have achieved?

- Potential strategy to induce users to switch from diesel power to grid power
- Stability; uncertainty; transient performance.

What are open to be done?

- Implementation of the proposed control and optimization on large-scale grids
- Hybrid systems modeling and control with boolean networks and continuous dynamics

Thanks!

zhubing@buaa.edu.cn