2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

An Optimization-Based Network Partitioning Method Considering Local Controllability: Application to Water Distribution Networks

Alireza Arastou, Ye Wang, and Erik Weyer

Abstract— **Complex networks pose significant challenges for design of control systems. This paper proposes a partitioning method for large-scale networks to be employed for use in decentralized and distributed control. The paper is divided into two parts. In the first part, a new formulation of the partitioning problem considering both computational and communication costs associated with control is established, and the controllability of the subsystems are also taken into account. In the second part, an effective algorithm is developed to find the solution to the network decomposition problem. The proposed approach is illustrated on a water distribution system benchmark.**

I. INTRODUCTION

Control of large-scale networks, such as water networks, power grids, and traffic systems are attracting increased interest since they are critical infrastructure [1], [2]. One option is to employ well-known control methods, such as linear quadratic regulator (LQR) or model predictive control (MPC), to design a centralized controller for the whole network [3]. However, dependence on a centralized controller makes the network prone to single-point failures [4]. It also leads to high computational costs and sensitive to failures associated with measurement collection and transmission of command signals to actuators [5]. To circumvent these problems, the use of decentralized and distributed control systems have gathered momentum. The general idea is to partition the system into smaller subsystems. Each subsystem is operated by a local controller, and the overall control system is the combined actions of these local controllers [6], [7]. Partitioning of the network into subsystems is the first step in the design of decentralized and distributed controllers.

Network decomposition using graph partitioning algorithms has been deployed in [8]–[14]. The underlying idea is that a dynamical system can be represented as a directed graph with states and inputs represented as vertices, and weighted edges representing dependencies between states and inputs. Once the graph of the network is derived, graph clustering approaches can be employed to evaluate subgraphs based on the desired objectives, e.g. finding subsystems with minimum couplings.

Although the aforementioned references have shown the effectiveness of their proposed methods on diverse case studies, there are some remaining challenges in the network decomposition problem. Firstly, the controllability of subsystems should be guaranteed for design of decentralized control systems to make sure all states can be steered to desired values using local controllers. Secondly, the partitioning objectives in terms of minimizing communication and computational costs of the control systems to be designed should be taken into account. Finally, most of the above cited works, such as [9]–[12], have only considered a fixed number of partitions. However, the number of partitions can have a significant impact on the resulting control performance.

The contributions of this paper are summarized as follows:

- A new formulation of the network partitioning problem, considering the number of partitions as a decision variable;
- Several new candidate objective functions for minimization of communication and computational costs;
- The use of controllability of subsystems as a constraint in the partitioning algorithm;
- An efficient algorithm for solving the partitioning problem.

The proposed approach is applied to the Richmond water distribution network and the results show the effectiveness of the proposed algorithm in terms of partitioning cost and control performance.

The paper is organized as follows: Section II gives the problem statement. The proposed objectives for the partitioning problem are discussed in Section III. Section IV develops an algorithm to solve the network decomposition problem. Simulation results are shown in Section V and conclusions are given in Section VI.

Notation: Sets are indicated by calligraphic upper case letters, e.g., $\mathcal{V}$. The square of weighted Euclidean norm is denoted by $\|.\|_R^2$, i.e, $\|x\|_R^2 = x^T R x$ for all $x \in \mathbb{R}^n$. Graphs are denoted as $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of vertices and edges, respectively. If $\mathcal{V}$ contains n vertices, the $n \times n$ matrix of binary values in which (i, j) entry is 1 if $e_{ij} \in \mathcal{E}$ and 0 otherwise, is the adjacency matrix of the graph. The neighborhood set of a vertex $\theta \in \mathcal{V}$ is denoted by $\mathcal{N}_\theta$ and contains the vertices that are connected to θ, i.e., $\mathcal{N}_\theta = \{v | e_{\theta v} \in \mathcal{E}\} \cup \{v | e_{v\theta} \in \mathcal{E}\}$. Finally, for an edge e_{ij} in a directed graph, $i \in \mathcal{V}$ is called the starting vertex and $j \in \mathcal{V}$ is be the end vertex.

II. PROBLEM STATEMENT

The problem of decomposing a large-scale network into multiple subsystems is a crucial step in the design of a decentralized or distributed control system. To this end, a general mathematical model for water networks as a case study for this paper is derived, and the partitioning problem is described in this section. Although the proposed partitioning

Alireza Arastou, Ye Wang, and Erik Weyer are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Melbourne, Victoria, 3010, Australia, E-mails: aarastou@student.unimelb.edu.au, ye.wang1@unimelb.edu.au, ewey@unimelb.edu.au

algorithm is general, we have restricted our attention to water networks.

A. Mathematical Model of Water Distribution Networks

A water distribution network comprises tanks, actuators (pumps and control valves), nodes, pipes, and demand sectors, which are modeled in the following parts. The overall water network model is the combination of the models of the different components [15]–[17].

A control-oriented model for water networks can be established by applying a volume-balance to the tanks and a flow-balance to nodes [10], [16]. The volume-balance for the l-th tank is given by:

$$x_l(k+1) = x_l(k) + \frac{\Delta t}{S_l}(q_{i,l}(k) - q_{o,l}(k)), \tag{1}$$

where x_l denotes the water level, $q_{i,l}$ and $q_{o,l}$ are in- and out-flows, and Δt is the sampling time, and S_l is the area of tank l. Similarly, the flow-balance for a node states that the sum of the inflows is equal to the sum of the outflows:

$$\sum_{w=1}^{n_{i,l}} q^l_{i,w}(k) = \sum_{j=1}^{n_{o,l}} q^l_{o,j}(k), \tag{2}$$

where $n_{i,l}$ and $n_{o,l}$ are the number of in- and out-flows for the l-th node.

The flows in (1) and (2) can be divided into two types: flows that are set by actuators denoted by u and flows that represent demands denoted by d. The overall state-space model can be constructed from (1) and (2):

$$x(k+1) = Ax(k) + B_u u(k) + B_d d(k), \tag{3a}$$
$$0 = E_u u(k) + E_d d(k), \tag{3b}$$

where $x \in \mathbb{R}^n$ are states representing water levels in tanks, $u \in \mathbb{R}^m$ are the water flows through pumps and control valves, while $d \in \mathbb{R}^p$ are water demands. A, B_u, B_d, E_u, E_d are matrices of appropriate dimensions.

The water levels and manipulated flows through actuators must satisfy physical limitations set by the tank capacity and actuator characteristics:

$$\underline{x} \le x \le \overline{x}, \qquad \underline{u} \le u \le \overline{u}, \tag{4}$$

where $\underline{x}$ and $\overline{x}$, are lower and upper bounds of water level, and $\underline{u}$ and $\overline{u}$ are lower and upper bounds of manipulated flow through actuators. (The inequalities in (4) hold element wise.)

The derived model in (3) can be represented as a directed graph. To this end, the states, nodes and inputs are considered as graph vertices, and the edges are evaluated using the state-space representation matrices in (3).

Example 1: Fig. 1 depicts the graph representation of a system with two states and one node (denoted by N_1), and the matrices in (3) given by $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $B_u = \begin{bmatrix} 0.5 & 1 & 0 \\ 0 & 3 & 0 \end{bmatrix}$, $B_d = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$, $E_u = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$, $E_d = \begin{bmatrix} 0 & -1 \end{bmatrix}$. Inputs and demands are shown in brown, states in yellow, and the node in blue in Fig. 1.

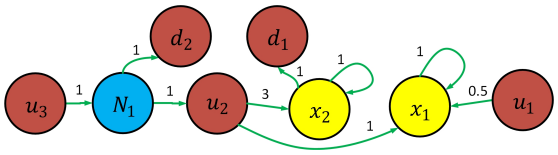

Fig. 1. Graph representation of the state-space model in Example 1.

B. Network Decomposition Problem

For large-scale water networks, the model in (3) and (4) can have many states, e.g., the Barcelona drinking water network model has 17 states [10]. Models of networks like power grids and traffic systems also have a large number of states [18], [19]. Therefore, deploying decentralized or distributed control systems may lead to a reduced computational burden, an increased reliability, and enhanced flexibility against network changes.

A typical objective is to minimize the couplings among subsystems to avoid excessive communications. Thus, it can be implied that the number of partitions should be reduced to a minimum. However, the subsystems should be sufficiently small to overcome problems with a centralized controller structure. Providing a trade-off between the computational and the communication burden is a challenging problem. Moreover, the controllability of the subsystems is crucial for decentralized control.

This paper consider the following optimization-based partitioning problem:

Problem 1: Consider a network as a directed graph with set of vertices, $\mathcal{V}$. The partitioning problem consists of finding the number of subsets P and a partitioning $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_P$ of $\mathcal{V}$ such that the following criterion is minimised subject to the given constraints.

$$\min_{P\in\{1,2,\ldots,m\}} \min_{\mathcal{V}_1,\mathcal{V}_2,\ldots,\mathcal{V}_P} \sum_{i=1}^{P} w_1 f_{cm}(\mathcal{V}_i) + w_2 f_{cp}(\mathcal{V}_i), \tag{5a}$$

$$\text{subject to:} \quad \bigcup_{i=1}^{P} \mathcal{V}_i = \mathcal{V}, \qquad \mathcal{V}_i \cap \mathcal{V}_j = \emptyset, \tag{5b}$$

$$\operatorname{rank}(C_i) = n_i, \quad i = 1, 2, \ldots, P, \tag{5c}$$

The partitioning cost function can be defined as follows:

$$f_t(\mathcal{V}) = \sum_{i=1}^{P} w_1 f_{cm}(\mathcal{V}_i) + w_2 f_{cp}(\mathcal{V}_i), \tag{6}$$

where f_{cm}, and f_{cp} are functions related to communication and computational costs, respectively. m is the total number of inputs, w_1, and w_2 are weights, n_i is the number of states in the i-th subsystem, and (5c) guarantees the controllability of the subsystems by requiring local controllability matrices

$$C_i = C(A_i, B_{u,i}) = [B_{u,i} \;\; A_i B_{u,i}, \ldots, A_i^{n_i-1} B_{u,i}], \tag{7}$$

to be full rank. A_i and $B_{u,i}$ are the state-space representation matrices of the i-th subsystem corresponding to $\mathcal{V}_i$ and $\mathcal{E}_i$.

In the next section, we will discuss several options for candidate functions for the communication and computational costs.

III. Candidate Functions for Communication and Computational Costs

In this section, several candidate functions are proposed to capture the computational and communication costs.

A. Communication Cost

1) Number of cuts: Number of cuts is defined as the minimum number of edges that must be removed from a graph such that a subsystem becomes isolated. The number of communication links between controllers can be represented by the number of cuts in the graph. As described in Section II-B, it is preferred to reduce the number of couplings among subsystems, so that the number of communications among local controllers in a distributed control strategy is minimized. Moreover, the disturbances caused by control actions in another subsystem should be small.

This goal can be fulfilled by minimizing the number of cuts in the graph [8] considering the weight of the edges. This idea can be further generalized to include the weights of the edges in a directed graph as follows:

$$f_{ct}(\mathcal{V}_l) = \sum_{i \in \mathcal{V}_l} \sum_{j \in \mathcal{V} \setminus \mathcal{V}_l} a_{ij} + a_{ji}, \tag{8}$$

where a_{ij} and a_{ji} are the weights of the links that connect the l-th subsystem to its neighbors. The weights are computed using the model representation in (3), as depicted in Fig. 1.

2) Total communication delay: From a control perspective, a delay can degrade the control performance. Fig. 2 illustrates the necessity for considering the communication delay in the partitioning procedure. The number of cuts are equal for both partitioning in Fig. 2.

Assume that there is only one low bandwidth communication channel between each pair of controllers such that the controllers can transmit one signal (input) at a time with time delay T_d. Assuming that parallel communication between different subsystems can take place, the total communication delay equals the maximum couplings between each pair of subsystems. This value for the partitioning in (2a) and (2b) are equal to $2T_d$ and $3T_d$, respectively. The total delay for the aforementioned communication protocol can be expressed as:

$$\begin{aligned} f_{cd}(\mathcal{V}_l) = \max_{\mathcal{V}_t} \sum_{i \in \mathcal{V}_l} \sum_{j \in \mathcal{V}_t} (\bar{a}_{ij} + \bar{a}_{ji}) T_d, \\ \forall\, l, t \in \{1, 2, \ldots, P\}, \quad t \neq l, \end{aligned} \tag{9}$$

where $\bar{a}_{ij}$ equals one if a_{ij} is nonzero and zero otherwise. In order to provide a trade-off between the aforementioned costs, a combination of (8) and (9) with suitable weights is proposed as the communication cost:

$$f_{cm}(\mathcal{V}_l) = \alpha_1 f_{ct}(\mathcal{V}_l) + \alpha_2 f_{cd}(\mathcal{V}_l), \tag{10}$$

If the communication channels between local controllers have a high bandwidth and can transmit all packages at the same time, the delay will not be an issue, and in that case α_2 can be chosen small or set to zero.

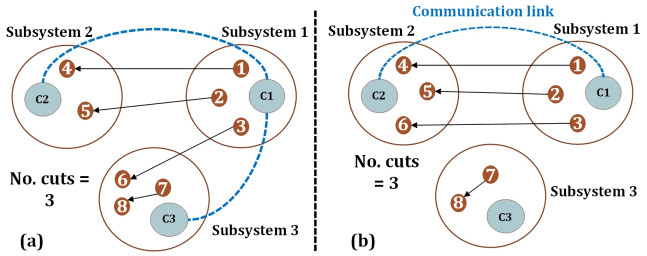

Fig. 2. Partitions with equal number of cuts and different total delay.

B. Computational Cost

Assume a control strategy such as MPC is selected to design local controllers. The computational complexity of solving the associated optimization problems relies on several factors, such as the number of decision variables, states, and the algorithm used for finding the solution. The latter is usually expressed using big-O notation. For example, the computational complexity for solving an MPC problem using quadratic programming, with n states, m inputs, and a prediction horizon of N by utilizing an interior point method is $O(N^3(n+m)^3)$ [20]. Hence, the computational cost function can be expressed as:

$$f_{cp}(\mathcal{V}_l) = O(g(n_l, m_l, T)), \tag{11}$$

where the function $g(.)$ depends on the algorithm for solving the problem, and n_l and m_l are number of states and inputs in the l-th subsystem. T denotes variables associated with the controller, such as the prediction horizon in the case of MPC.

The following proposition sheds light on the relationship between having a balanced number of states and inputs at each subsystem and the computational cost.

Proposition 1: If $g(n_l, m_l, T) = f(T)(n_l^{k_1} + m_l^{k_2})$ in (11) with f a non-negative function, and k_1 and k_2 are positive integers. Then, assigning a balanced number of states and inputs to each subsystem leads to a minimum computational cost.

Proof: The minimization problem is:

$$\min_{\substack{n_1,\ldots,n_{P-1} \\ m_1,\ldots,m_{P-1}}} \sum_{i=1}^{P} f(T)(n_i^{k_1} + m_i^{k_2}), \tag{12a}$$

$$\text{subject to : } \sum_{i=1}^{P} n_i = n, \qquad \sum_{i=1}^{P} m_i = m. \tag{12b}$$

By introducing Lagrange multipliers λ and μ, we obtain:

$$L = \sum_{i=1}^{P} f(T)(n_i^{k_1} + m_i^{k_2}) - \lambda(\sum_{i=1}^{P} n_i - n) - \mu(\sum_{i=1}^{P} m_i - m), \tag{13}$$

By taking partial derivatives of (13), and solving $\frac{\partial L}{\partial n_i} = 0$ and $\frac{\partial L}{\partial m_i} = 0$, we have:

$$n_i = \sqrt[k_1 - 1]{\frac{\lambda}{f(T)k_1}}, \qquad m_i = \sqrt[k_2 - 1]{\frac{\mu}{f(T)k_2}}. \tag{14}$$

Since the right hand sides in (14) do not depend on i, the number of states for all subsystems must be equal, and the same holds for the number of inputs. From (12b) and (14), we have:

$$n_i^* = \frac{n}{P}, \qquad m_i^* = \frac{m}{P}. \tag{15}$$

■

Based on *Proposition 1*, the following imbalance cost, which was employed in [9], [11], [12] as well, is proposed to be considered as the computational cost function:

$$f_{cp}(\mathcal{V}_l) = (n_l - \frac{n}{P})^2 + (m_l - \frac{m}{P})^2. \tag{16}$$

IV. THE PROPOSED PARTITIONING ALGORITHM

In this section an algorithm is developed to solve the network partitioning problem in (5). The proposed approach is based on the method in [12]. The proposed algorithm contains an initialization part, a selection of an initial partitioning, and the refinement stage.

A. Initialization

It is desirable to provide the algorithm with a warm start to enhance the convergence speed. To this aim, the vertices are aggregated such that the controllability of some states are guaranteed during the partitioning procedure. Based on [21], the following assignment rule is applied:

Assignment rule: For a network represented as a directed graph and with decoupled states, if a state (a water level in a tank) is connected to only one input (an actuator), their corresponding vertices are merged to guarantee they are assigned to the same partition.

The adjacency matrix is correspondingly modified, and the modified adjacency matrix denoted by A_r, will be used. This step also includes selecting weights in (5a), and specifying the number of states and inputs, i.e., n and m.

B. Initial Partitioning

The main idea behind the initial partitioning step is to put the vertices with strong connectivity in the same partition. Thus, the optimization problem is not addressed in this stage. Hence, the first P inputs with highest degree centrality, i.e., highest number of edges connected to a vertex, are assigned as the centers of each partition. Starting from subsystem 1, other vertices that have not been assigned to any partition yet, and have the highest number of couplings to the evaluated subsystem, are assigned to it. (If some vertices have the same maximum number of couplings to a subsystem, we choose the assigned vertex randomly from the maximizers). After a vertex is assigned to a subsystem, we proceed to the next subsystem. The procedure is repeated until all vertices are assigned to a subsystem.

Furthermore, the initial partitioning should be controllable. This is achieved in this stage by ensuring each state is connected to at least one input.

C. Refinement Stage

The output of previous steps is a decomposition of the network into P controllable partitions. The initial partitions are next modified so that the total cost function in (5a) is reduced and the constraints in (5b)-(5c) are fulfilled.

The proposed algorithm is given in Algorithm 1. The refinement process starts with the subsystem with highest local cost. The possible modification for this subsystem is to move a single vertex that has coupling to other partitions to the neighboring partitions. These vertices are put in a set called $\hat{\mathcal{V}}_l$. Moving a vertex in $\hat{\mathcal{V}}_l$ to the neighboring subsystems would give a new value for the cost function in (5a). The vertex which gives the largest reduction in the cost function is selected and transferred to the neighboring partitions. After a vertex has been assigned to a new subsystem, we repeat sorting process based on highest local costs. The refinement algorithm will be repeated until a stopping criterion is satisfied.

Algorithm 1 Proposed partitioning method

Initialization: $n, m, w_1, w_2, \alpha_1, \alpha_2$, and A_r
for $P = 1$ to m **do**
Inputs: Generate an initial partitioning using the procedure in Section IV.B. Call it $\mathcal{V}^o$. Set $r = 1, c = 0, j = 1$
Sort each subsystem in $\mathcal{V}^o$ according to local cost in descending order into an array of subsystems V_{so}
  while $c \leq P$ **do**
    $\mathcal{V}_l = V_{so}(j)$
    $\hat{\mathcal{V}}_l = \{\theta_w | e_{wz} \in \mathcal{E}, \theta_w \in \mathcal{V}_l, \theta_z \in \mathcal{V} \backslash \mathcal{V}_l\}$
    for each $\theta \in \hat{\mathcal{V}}_l$ **do**
      $\mathcal{N}_{\hat{\mathcal{V}}_l} = \{\mathcal{V}_h \in \mathcal{V}^o | \mathcal{N}_\theta \cap \mathcal{V}_h \neq \emptyset\} \backslash \mathcal{V}_l$
      $\mathcal{V}_l^n = \mathcal{V}_l \backslash \{\theta\}$
      for each $\mathcal{V}_t \in \mathcal{N}_{\hat{\mathcal{V}}_l}$ **do**
        $\mathcal{V}_t^n = \mathcal{V}_t \cup \{\theta\}$
        $\mathcal{V}^n = \{\mathcal{V}_l^n, \mathcal{V}_t^n, \mathcal{V}^o \backslash \{\mathcal{V}_l, \mathcal{V}_t\}\}$
        $\Delta f_t = f_t(\mathcal{V}^o) - f_t(\mathcal{V}^n)$ using (6)
        if $\Delta f_t > 0$ **then**
          $F_d(r) = \Delta f_t, V'(r) = \mathcal{V}^n$
          $r = r + 1$
        end if
      end for $\mathcal{V}_t$
    end for θ
    if $r > 1$ **then**
      Find r that maximizes $F_d(r)$ and gives full rank controllability matrices for $V'(r)$.
      $\mathcal{V}^o = V'(r), c = 0$, Sort $\mathcal{V}^o$ and store it in V_{so}, $j = 0$
    else
      $c = c + 1$
    end if
    Reset F_d, V', and r
    $j = j + 1$
  end while c
  $V_p(P) = \mathcal{V}^o,\ f_p(P) = f_t(\mathcal{V}^o)$
end for P
Output: P^* and $V_p(P^*)$, which give minimum $f_p(P)$

Note that the total cost function for each partitioning is non-increasing using the proposed algorithm, since the partitioning is modified only if the total cost is reduced. The refinement stage is repeated until no improvement in the partitioning is seen or a maximum number of iterations is not exceeded in the case of large scale network. A convergence proof for a similar algorithm is given in [12].

Lastly, the optimization problem defined in (5) proposes a min-min formulation to find the optimal number of partitions, P, as well. Accordingly, the initial partitioning and refinement steps are repeated for all values of P to find the subsystems, which minimizes total cost. The upper bound on the number of partitions is m to guarantee that each each subsystem contains at least one input.

V. Case Study: Richmond Water Network

The Richmond water network is part of the Yorkshire water supply area in U.K. ([22] and [23]). A schematic representation of this network called "Richmond skeleton" [23], is shown in Fig. 3. The corresponding vertices for the pumps in 1A, 2A, and 3A have been merged and labeled as u_A.

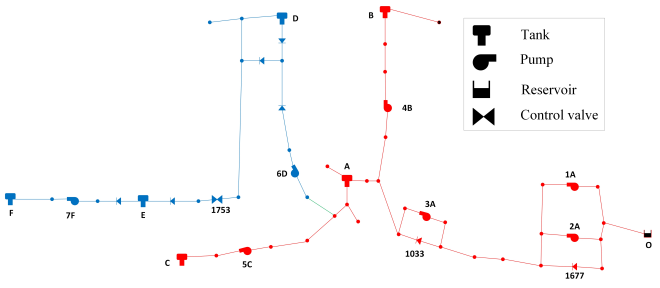

Fig. 3. The partitioning results for the Richmond skeleton water network.

1) Network Partitioning Results: For this case study, we consider $n = 6, m = 6, w_1 = 1, w_2 = 1, \alpha_1 = 2, \alpha_2 = 0$. Moreover, (10) and (16) were selected as communication and computational costs. The obtained partitions using the proposed partitioning algorithm are indicated in Fig. 3 with different colors. The values of the total cost for different number of partitions are shown in Table I. From Table I, it is evident that $P = 2$ gives the minimum value.

TABLE I
Partitioning cost for Richmond water network.

P	2	3	4	5	6
Partitioning Cost	2	8	8	9.6	10

2) Comparison of the control performance using the obtained partitions: A comparison of the control performance for different number of subsystems using a distributed cooperative economic MPC strategy is presented in this part. The control objectives are minimization of energy cost due to pumping and smoothness of control actions [16]. Following [24], each controller solves the following problem at each iteration, r, until a stopping criterion is satisfied:

$$[u_l^*(k), \ldots, u_l^*(k+N-1)] = \arg \min_{u_l(k),\ldots,u_l(k+N-1)} \sum_{j=0}^{N-1} J(u_l(k+j), u_{-l}(k+j)), \tag{17a}$$

subject to (3) and (4), for $j = 0, \ldots, N-1$

$$u_{-l}^r(k+j) = u_{-l}^{r-1}(k+j), \tag{17b}$$

$$u_l^r(k+j) = \beta_l u_l^{r-1}(k+j) + (1-\beta_l) u_l^*(k+j). \tag{17c}$$

with

$$\mathbf{u}(k) = [u_l(k), u_{-l}(k)]^T \tag{18a}$$

$$J(\mathbf{u}(k)) = \sum_{l=1}^{P} \alpha(k) u_l(k) + \|\Delta u_l(k)\|_{R_l}^2, \tag{18b}$$

$$\Delta u_l(k) = u_l(k) - u_l(k-1), \tag{18c}$$

stopping criterion:

$$|J(\mathbf{u}^r(k)) - J(\mathbf{u}_c(k))| \leq \epsilon \quad \text{or} \quad r \geq \bar{r} \tag{19}$$

where $u_l(k)$ and $u_{-l}(k)$ denote the inputs in l-th and other subsystems, respectively. $\mathbf{u}_c(k)$ is the solution of the centralized controller, $\mathbf{u}^r$ is the input vector in (18a) at r-th iteration, $\bar{r}$ is the upper bound for number of iterations, ϵ shows the desired accuracy, N is the prediction horizon, $\alpha(k)$ denotes the electricity tariff, R_l is weight for input smoothness in the l-th subsystem. After each iteration, the controllers communicate their solutions to each other. The control actions by the other subsystems act as disturbances on the subsystem under consideration in (17b). Moreover, the input trajectory at iteration r, $u_l^r(k)$, is computed in (17c) using the solution in the previous iteration, the optimal solution found by (17a), and β_l. In this case study, $J(\mathbf{u}_c(k)) = 70298$ $\epsilon = 2000$, $\bar{r} = 24$, $N = 24$ hours, the total simulation period was 72 hours, $R_l = 10 I_{m_l}$, $\Delta t = 5$ minutes, $\overline{x} = [3.37, 3.65, 2, 2.11, 2.19]^T$, $\underline{x} = 0.5\overline{x}$, $\overline{u} = 50$, $\underline{u} = 0$, and $\beta_l = 0.5$. Note that we used the stopping condition in (19),

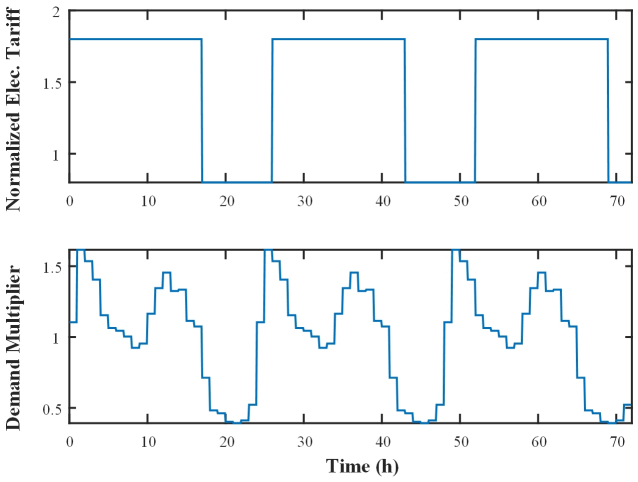

Fig. 4. Normalized electricity tariff and demand multiplier.

since $J(\mathbf{u}_c)$ was easy to obtain. In general, the centralized solution is unknown, and $|J(\mathbf{u}^r(k)) - J(\mathbf{u}^{r-1}(k))| \leq \epsilon$ can be employed as the termination condition. Moreover, electricity tariff and demand multiplier are shown in Fig. 4. The demands are given by $d(k) = m(k)\bar{d}$ where $m(k)$ is the demand multiplier shown in Fig. 4. The same demand multiplier was assumed for all demands. The demand multiplier has an average of 1, and hence the average demand is $\bar{d}$. The simulation results are given in Table II.

TABLE II
COMPARISON OF CONTROL PERFORMANCE FOR DIFFERENT PARTITIONS.

P	$T_r(s)$	I_r	$T_s(s)$	C_e	C_s	Total operating cost
2	0.38	2.25	0.86	63584	7233	70817
3	0.41	3.69	1.51	63752	7364	71116
4	0.30	9.34	2.80	63759	7488	71247
5	0.29	15.4	4.46	64035	7691	71726
6	0.09	23.6	2.13	66248	7933	74181

T_r is the average simulation time per iteration for the most computationally demanding subsystem, I_r is the average number of iterations per time step, T_s is the average simulation time per time step for the most computationally demanding subsystem, C_e is the energy cost, and C_s is the input smoothness cost.

From Table II, it is evident that the number of iterations to find the solution increases as the number of subsystems increases because more subsystems needs to communicate their results to each other. It can be seen from Table II that $P = 2$ led to the best control performance in terms of simulation time and total control cost. This statement is compatible with the partitioning result in Table I. Furthermore, it is evident that T_r for $P = 6$ is decreased considerably in comparison to $P = 5$. The main reason is that the largest subsystem in the case of $P = 5$ contains two inputs and two states, while it includes one state and one input for $P = 6$.

VI. CONCLUSION

A novel method for network decomposition has been proposed. The aim of the approach is find a partition such that the computational costs for local controllers and the communication costs between local controllers is minimized. To achieve this, a graph representation of the network was used and several costs functions were proposed, and controllability constraints were imposed. An efficient method was also proposed for solving the obtained partitioning problem.

The proposed approach was applied to a case study, and simulation results comprising the obtained partitions and control performance for different number of partitions were presented. The algorithm successfully found the partitioning that led to the best control performance, demonstrating its suitability.

REFERENCES

[1] A. Bemporad, M. Heemels, M. Johansson, *et al.*, *Networked control systems*, vol. 406. Springer, 2010.

[2] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, vol. 11, no. 10, pp. 1099–1111, 2003.

[3] J. B. Rawlings and B. T. Stewart, "Coordinating multiple optimization-based controllers: New opportunities and challenges," *Journal of Process Control*, vol. 18, no. 9, pp. 839–845, 2008.

[4] A. Bidram, F. L. Lewis, and A. Davoudi, "Distributed control systems for small-scale power networks: Using multiagent cooperative control theory," *IEEE Control Systems Magazine*, vol. 34, no. 6, pp. 56–77, 2014.

[5] J. M. Maestre and R. R. Negenborn, eds., *Distributed model predictive control made easy*, vol. 69. Springer, 2014.

[6] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*, vol. 2. Nob Hill Publishing Madison, WI, 2017.

[7] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[8] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz, *Recent Advances in Graph Partitioning*, pp. 117–158. Cham: Springer International Publishing, 2016.

[9] J. Barreiro-Gomez and J. Barreiro-Gomez, "Partitioning for large-scale systems: Sequential dmpc design," *The Role of Population Games in the Design of Optimization-Based Controllers: A Large-scale Insight*, pp. 163–178, 2019.

[10] C. Ocampo-Martinez, D. Barcelli, V. Puig, and A. Bemporad, "Hierarchical and decentralised model predictive control of drinking water networks: Application to barcelona case study," *IET control theory & applications*, vol. 6, no. 1, pp. 62–71, 2012.

[11] J. Barreiro-Gomez, C. Ocampo-Martinez, and N. Quijano, "Time-varying partitioning for predictive control design: Density-games approach," *Journal of Process Control*, vol. 75, pp. 1–14, 2019.

[12] W. Ananduta, T. Pippia, C. Ocampo-Martinez, J. Sijs, and B. De Schutter, "Online partitioning method for decentralized control of linear switching large-scale systems," *Journal of the Franklin Institute*, vol. 356, no. 6, pp. 3290–3313, 2019.

[13] W. Tang and P. Daoutidis, "Network decomposition for distributed control through community detection in input–output bipartite graphs," *Journal of Process Control*, vol. 64, pp. 7–14, 2018.

[14] R. R. Rocha, L. C. Oliveira-Lopes, and P. D. Christofides, "Partitioning for distributed model predictive control of nonlinear processes," *Chemical Engineering Research and Design*, vol. 139, pp. 116–135, 2018.

[15] M. A. Brdys and B. Ulanicki, *Operational control of water systems: structures, algorithms, and applications*. Prentice Hall, 1994.

[16] Y. Wang, V. Puig, and G. Cembrano, "Non-linear economic model predictive control of water distribution networks," *Journal of Process Control*, vol. 56, pp. 23–34, 2017.

[17] Y. Wang, K. Too Yok, W. Wu, A. R. Simpson, E. Weyer, and C. Manzie, "Minimizing pumping energy cost in real-time operations of water distribution systems using economic model predictive control," *Journal of Water Resources Planning and Management*, vol. 147, no. 7, p. 04021042, 2021.

[18] F. Dörfler, M. R. Jovanović, M. Chertkov, and F. Bullo, "Sparsity-promoting optimal wide-area control of power networks," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2281–2291, 2014.

[19] H. Zhao, X. Dai, Q. Zhang, and J. Ding, "Robust event-triggered model predictive control for multiple high-speed trains with switching topologies," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 4700–4710, 2020.

[20] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2009.

[21] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[22] J. E. Van Zyl, D. A. Savic, and G. A. Walters, "Operational optimization of water distribution systems using a hybrid genetic algorithm," *Journal of Water Resources Planning and Management*, vol. 130, no. 2, pp. 160–170, 2004.

[23] U. of Exeter, "Richmond network." https://engineering.exeter.ac.uk/research/cws/resources/benchmarks/, 2000. Accessed February 21, 2023.

[24] P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers & Chemical Engineering*, vol. 51, pp. 21–41, 2013.