A Game Theoretical Randomized Method for Large-Scale Systems Partitioning

FRANCISCO JAVIER MUROS [1], JOSÉ MARÍA MAESTRE [1], (Senior Member, IEEE), CARLOS OCAMPO-MARTINEZ [2], (Senior Member, IEEE), ENCARNACIÓN ALGABA [3], and EDUARDO F. CAMACHO [1], (Fellow, IEEE)

[1]Department of Systems and Automation Engineering, University of Seville, Spain (e-mail: {franmuros,pepemaestre,efcamacho}@us.es)

[2]Automatic Control Department, Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain (e-mail: cocampo@iri.upc.edu)

[3]Department of Applied Mathematics II, University of Seville, Spain (e-mail: ealgaba@us.es)

Corresponding author: Francisco Javier Muros (e-mail: franmuros@us.es)

This work has been partially supported by the European Union Project FP7-ICT-ICT-2013.3.4-611281 (DYMASOS), and the MINECO-Spain Projects DPI2016-78338-R (CONFIGURA), DPI2017-86918-R, ECO2015-68856-P and DPI2016-76493-C3-3-R (DEOCS).

ABSTRACT In this paper, a game theory-based partitioning algorithm for large-scale systems (LSS) is proposed. More specifically, a game over nodes is introduced in a model predictive control framework. The Shapley value of this game is used to rank the communication links of the control network based on their impact on the overall system performance. A randomized method to estimate the Shapley value of each node and also an efficient redistribution of the resulting value to the links involved are considered to relieve the combinatorial explosion issues related to LSS. Once the partitioning solution is obtained, a sensitivity analysis is proposed to give a measure of its performance. Likewise, a greedy fine tuning procedure is considered to increase the optimality of the partitioning results. The full Barcelona drinking water network (DWN) is analyzed as a real LSS case study showing the effectiveness of the proposed approach in comparison with other partitioning schemes available in the literature.

INDEX TERMS Coalitional control, cooperative game theory, system partitioning, randomized methods, Shapley value, large-scale systems (LSS), drinking water networks (DWN).

I. INTRODUCTION

MODEL predictive control (MPC) has evolved considerably over the last decades. It designates an ample range of control methods that make explicit use of a model of the process to obtain the control signal by minimizing an objective function [1]. Its flexibility – for any type of model can be used – and the ease of dealing with constraints and dead times are well-known advantages of this methodology. In particular, this paper focuses on distributed MPC (DMPC) schemes, where the overall control problem is divided into smaller pieces assigned to local controllers or *agents*, which are able to communicate among them. Typical features of these approaches, such as scalability, modularity, and the capacity of controlling large-scale systems (LSS) [2], will be very welcome in this work.

Each agent involved in a distributed scheme has partial system information and communicates with neighbors for the sake of coordination [3]. Typically, the *partition* of the overall system is assumed to be given before the DMPC strategy is applied, and it is calculated during the system modeling, based on physical insight, experience or intuition, and other methods available in the literature, where the starting point is commonly associated with the seminal work of Siljak [4]. Recently, many partitioning schemes have appeared, based on graph theory [5]–[8], states/inputs estimation [9], social network algorithms [10], genetic algorithms [11], or *PageRank* [12]. In any case, no single partitioning strategy is the best fit for all situations. This way, specific partitioning techniques have been applied to real LSS case studies, e.g., in water systems [10], [13]–[15], power networks [16], [17], biological systems [18], integrated circuits [19], and urban traffic networks [20]. In fact, LSS might involve a big communication network implying the handling of large amount of data, which could yield in high computational costs. Therefore, performing the partition of the problem into smaller pieces is a natural solution for managing these networks.

In this work, the *coalitional control* framework, which can be summarized as a set of dynamic partitioning methods for networked control systems that seek for a trade-off between communication burden and control performance [21]–[26], is used to determine *static* neighborhoods that define the partitioning of an LSS. That is, despite its dynamic scope, the coalitional framework can be used to obtain *offline* system information, i.e., independent from the implementation of the control scheme. In this coalitional context, the well-known Shapley value, a cooperative game solution concept presented originally in [27], is used here to provide information regarding the relevance of the agents and the links involved in a communication network, following the ideas described in [21]–[23]. The specific contributions of this article with respect these earlier works are the following:

a) A game defined in the set of *agents* is proposed here, where the closed-loop stage cost of the system is accumulated along the simulation scenario. This is a difference with respect to previous works, were open-loop costs where used at each time step. This game is associated with the cost function of a coalitional MPC scheme, in the line of [23]. Note that the viewpoint is changed with respect to [21], [22], where a game with the players being the *links* was related to the cost function of a coalitional approach based on linear feedback gains.

b) The Shapley value, which gives an averaged contribution of each player into the game, is calculated here by using randomized methods [28], [29] satisfying a minimum bounded error requirement. The relevance of the links is obtained by an index, introduced in [23], which redistributes the Shapley value from the agents to the links.

c) In [21], [23], a very preliminar partitioning that uses few thresholds to classify the links as a function of the control performance is proposed. Here, the partitioning algorithm is drastically improved by introducing new parameters that balance the size of the communication components. Additionally, a new set that refers to expensive links among components is also introduced here to provide a proper definition of the network configuration after applying the partitioning method.

d) A sensitivity analysis that gives a measure of the partitioning control performance is included in this paper. Through this analysis, the proposed partitioning solution is compared with other schemes available in the literature [10], [13], and also with the centralized and decentralized configurations. Moreover, this analysis is recursively implemented in a *greedy* fashion [30], [31] to make a fine tuning of the partitioning approach.

Notice that the results reported in previous works were only suitable for academic small networks. Nevertheless, through the combination of a) and b) the computational and combinatorial explosion issues related to LSS are mitigated and it is possible to implement the new partitioning algorithm introduced in c) and d) in such networks. Moreover, the full Barcelona drinking water network (DWN), modeled in [13], [14], is chosen as a real LSS case study to assess the effectiveness of the proposed partitioning approach, which represents an additional contribution of this paper.

The remainder of this paper is organized as follows. In Section II, the problem setting is stated in a coalitional control framework. Next, in Section III, a game over nodes is considered, being the Shapley value utilized as a means of distributing the cost of the game among the agents. Likewise, a randomized method to estimate the Shapley value and a redistribution of this value to the set of links are proposed to deal with combinatorial explosion issues. In Section IV, a Shapley value-based partitioning algorithm is introduced. The partitioning performance is evaluated by means of a sensitivity analysis, whose information is also recursively used for a fine tuning of the proposed approach. In Section V, the Barcelona DWN is presented as the case study, with the corresponding partitioning results and comparisons with other approaches being presented in Section VI. Finally, conclusions are drawn in Section VII.

II. FOUNDATIONS OF COALITIONAL CONTROL

Consider the class of distributed linear systems composed of $\mathcal{N} = \{1, 2, \ldots, |\mathcal{N}|\}$ interconnected subsystems or agents. The dynamics of agent $i \in \mathcal{N}$ can be mathematically described, with $k \in \mathbb{Z}^+$, as

$$\begin{aligned}\mathbf{x}_i(k+1) &= \mathbf{A}_{ii}\mathbf{x}_i(k) + \mathbf{B}_{ii}\mathbf{u}_i(k) + \mathbf{w}_i(k),\\ \mathbf{w}_i(k) &= \sum_{j\neq i}[\mathbf{A}_{ij}\mathbf{x}_j(k) + \mathbf{B}_{ij}\mathbf{u}_j(k)] + \mathbf{B}_{\mathbf{p}_i}\mathbf{d}_i(k),\end{aligned} \tag{1}$$

where $\mathbf{x}_i(k) \in \mathbb{R}^{n_{\mathrm{x}_i}}$ is the state vector of agent i, $\mathbf{u}_i(k) \in \mathbb{R}^{n_{\mathrm{u}_i}}$ its corresponding input vector, and $\mathbf{w}_i(k) \in \mathbb{R}^{n_{\mathrm{x}_i}}$ the related disturbances, which can be either external to the whole system, denoted by $\mathbf{d}_i(k) \in \mathbb{R}^{n_{\mathrm{d}_i}}$, or be caused by the neighbors as well. Likewise, $\mathbf{A}_{ii} \in \mathbb{R}^{n_{\mathrm{x}_i}\times n_{\mathrm{x}_i}}$, $\mathbf{B}_{ii} \in \mathbb{R}^{n_{\mathrm{x}_i}\times n_{\mathrm{u}_i}}$, $\mathbf{A}_{ij} \in \mathbb{R}^{n_{\mathrm{x}_i}\times n_{\mathrm{x}_j}}$, $\mathbf{B}_{ij} \in \mathbb{R}^{n_{\mathrm{x}_i}\times n_{\mathrm{u}_j}}$ and $\mathbf{B}_{\mathbf{p}_i} \in \mathbb{R}^{n_{\mathrm{x}_i}\times n_{\mathrm{d}_i}}$ are system matrices of suitable dimensions. Both states and inputs are constrained into independent sets defined by a collection of linear inequalities, i.e.,

$$\mathbf{x}_i(k) \in \mathcal{X}_i,\ \mathcal{X}_i \subseteq \mathbb{R}^{n_{\mathrm{x}_i}}, \qquad \mathbf{u}_i(k) \in \mathcal{U}_i,\ \mathcal{U}_i \subseteq \mathbb{R}^{n_{\mathrm{u}_i}}. \tag{2}$$

A. NETWORKED COALITIONAL STRUCTURE

In standard coalitional control, the agents are merged *at each time instant* into several disjoint neighborhoods or communication components $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{n_\mathrm{c}}$, verifying $\bigcup_{s=1}^{n_\mathrm{c}} \mathcal{C}_s = \mathcal{N}$. Conversely, the goal in this paper is to use coalitional control to find a *time-independent* set denoted as $\mathcal{N}_\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{n_\mathrm{c}}\}$, i.e., $n_\mathrm{c} = |\mathcal{N}_\mathcal{C}|$, assuming that agents in $\mathcal{N}$ are initially connected by a network described by an undirected graph $(\mathcal{N}, \mathcal{E})$, where $\mathcal{E} = \mathcal{E}^{\mathcal{N}} = \mathcal{N} \times \mathcal{N}$ is the set of links corresponding to *all* feasible communication connections among the agents. Hence, the number of elements in both sets is, in the worst case, connected by [21], [23]

$$|\mathcal{E}| = \frac{|\mathcal{N}|(|\mathcal{N}|-1)}{2}, \tag{3}$$

which corresponds to the number of links of a complete network. This case, which is the most demanding in terms of problem size, will be assumed throughout the paper. Note that, under this premise, any group or coalition $\mathcal{C} \subseteq \mathcal{N}$ of agents is internally connected.

Each link $l \in \mathcal{E}$ can be classified according to its relevance from a control viewpoint. This way, it can be more profitable for the overall system performance to fix/disconnect some links permanently. In this work, the partitioning objective will focus on finding out those links. When a specific coalition $\mathcal{C}$ is formed, a model analogous to (1) is calculated at a coalition level, i.e.,

$$\begin{aligned} \mathbf{x}_{\mathcal{C}}(k+1) &= \mathbf{A}_{\mathcal{C}\mathcal{C}}\mathbf{x}_{\mathcal{C}}(k) + \mathbf{B}_{\mathcal{C}\mathcal{C}}\mathbf{u}_{\mathcal{C}}(k) + \mathbf{w}_{\mathcal{C}}(k), \\ \mathbf{w}_{\mathcal{C}}(k) &= \sum_{j \notin \mathcal{C}} [\mathbf{A}_{\mathcal{C}j}\mathbf{x}_j(k) + \mathbf{B}_{\mathcal{C}j}\mathbf{u}_j(k)] + \mathbf{B}_{\mathbf{p}_{\mathcal{C}}}\mathbf{d}_{\mathcal{C}}(k), \end{aligned} \tag{4}$$

with $\mathbf{x}_{\mathcal{C}}(k) = [\mathbf{x}_i(k)]_{i\in\mathcal{C}} \in \mathbb{R}^{n_{\mathrm{x}_\mathcal{C}}}$, $\mathbf{u}_{\mathcal{C}}(k) = [\mathbf{u}_i(k)]_{i\in\mathcal{C}} \in \mathbb{R}^{n_{\mathrm{u}_\mathcal{C}}}$ and $\mathbf{w}_{\mathcal{C}}(k) = [\mathbf{w}_i(k)]_{i\in\mathcal{C}} \in \mathbb{R}^{n_{\mathrm{w}_\mathcal{C}}}$, $\mathbf{d}_{\mathcal{C}}(k) = [\mathbf{d}_i(k)]_{i\in\mathcal{C}} \in \mathbb{R}^{n_{\mathrm{d}_\mathcal{C}}}$ being respectively the coalitional states, inputs, overall disturbances and external disturbances that aggregate the corresponding vectors, and $\mathbf{A}_{\mathcal{C}\mathcal{C}} \in \mathbb{R}^{n_{\mathrm{x}_\mathcal{C}} \times n_{\mathrm{x}_\mathcal{C}}}$, $\mathbf{B}_{\mathcal{C}\mathcal{C}} \in \mathbb{R}^{n_{\mathrm{x}_\mathcal{C}} \times n_{\mathrm{u}_\mathcal{C}}}$, $\mathbf{A}_{\mathcal{C}j} \in \mathbb{R}^{n_{\mathrm{x}_\mathcal{C}} \times n_{\mathrm{x}_j}}$, $\mathbf{B}_{\mathcal{C}j} \in \mathbb{R}^{n_{\mathrm{x}_\mathcal{C}} \times n_{\mathrm{u}_j}}$ and $\mathbf{B}_{\mathbf{p}_{\mathcal{C}}} \in \mathbb{R}^{n_{\mathrm{x}_\mathcal{C}} \times n_{\mathrm{d}_\mathcal{C}}}$ are obtained by aggregating the corresponding individual matrices. The coalitional constraints become

$$\begin{aligned} \mathbf{x}_{\mathcal{C}}(k) \in \mathcal{X}_{\mathcal{C}} \subseteq \mathbb{R}^{n_{\mathrm{x}_\mathcal{C}}}, \quad \mathcal{X}_{\mathcal{C}} = \textstyle\prod_{i\in\mathcal{C}} \mathcal{X}_i, \\ \mathbf{u}_{\mathcal{C}}(k) \in \mathcal{U}_{\mathcal{C}} \subseteq \mathbb{R}^{n_{\mathrm{u}_\mathcal{C}}}, \quad \mathcal{U}_{\mathcal{C}} = \textstyle\prod_{i\in\mathcal{C}} \mathcal{U}_i. \end{aligned} \tag{5}$$

Finally, from an overall centralized viewpoint, the system is described by

$$\mathbf{x}_{\mathcal{N}}(k+1) = \mathbf{A}_{\mathcal{N}}\mathbf{x}_{\mathcal{N}}(k) + \mathbf{B}_{\mathcal{N}}\mathbf{u}_{\mathcal{N}}(k) + \mathbf{B}_{\mathbf{p}_{\mathcal{N}}}\mathbf{d}_{\mathcal{N}}(k), \tag{6}$$

with $\mathbf{x}_{\mathcal{N}}(k) \in \mathbb{R}^{n_{\mathrm{x}_\mathcal{N}}}$, $\mathbf{u}_{\mathcal{N}}(k) \in \mathbb{R}^{n_{\mathrm{u}_\mathcal{N}}}$, $\mathbf{d}_{\mathcal{N}}(k) \in \mathbb{R}^{n_{\mathrm{d}_\mathcal{N}}}$ being, respectively, the overall state, input and disturbance vectors. Note that from a centralized viewpoint, $\mathbf{w}_{\mathcal{N}}(k)$ is only composed of external disturbances $\mathbf{B}_{\mathbf{p}_{\mathcal{N}}}\mathbf{d}_{\mathcal{N}}(k)$. The centralized constraints have the form

$$\begin{aligned} \mathbf{x}_{\mathcal{N}}(k) \in \mathcal{X}_{\mathcal{N}} \subseteq \mathbb{R}^{n_{\mathrm{x}_\mathcal{N}}}, \quad \mathcal{X}_{\mathcal{N}} = \textstyle\prod_{i\in\mathcal{N}} \mathcal{X}_i, \\ \mathbf{u}_{\mathcal{N}}(k) \in \mathcal{U}_{\mathcal{N}} \subseteq \mathbb{R}^{n_{\mathrm{u}_\mathcal{N}}}, \quad \mathcal{U}_{\mathcal{N}} = \textstyle\prod_{i\in\mathcal{N}} \mathcal{U}_i. \end{aligned} \tag{7}$$

B. CONTROL OBJECTIVE

Under an MPC framework, the goal of each coalitional controller $\mathcal{C} \subseteq \mathcal{N}$ is to drive a sequence of future states over a prediction horizon N_{p}, that is, $\mathbf{X}_{\mathcal{C}}(k+1 : k+N_{\mathrm{p}}) = \{\mathbf{x}_{\mathcal{C}}(k+1), \ldots, \mathbf{x}_{\mathcal{C}}(k+N_{\mathrm{p}})\}$, by using the most appropriate control sequence. To this end, the controller solves the following open-loop finite-horizon optimization problem at each time instant k:

$$\mathbf{U}^*_{\mathcal{C}}(k : k+N_{\mathrm{p}}-1) = \arg \min_{\mathbf{U}_{\mathcal{C}}(k:k+N_{\mathrm{p}}-1)} \sum_{r=0}^{N_{\mathrm{p}}-1} \ell_{\mathcal{C}}\big(\mathbf{x}_{\mathcal{C}}(k+r+1), \mathbf{u}_{\mathcal{C}}(k+r)\big), \tag{8}$$

subject to (4), (5), the aggregate forecast of the expected local disturbances $\hat{\mathbf{W}}_{\mathcal{C}}(k : k+N_{\mathrm{p}}-1) = \{\hat{\mathbf{w}}_{\mathcal{C}}(k), \ldots, \hat{\mathbf{w}}_{\mathcal{C}}(k+N_{\mathrm{p}}-1)\}$, and a measured coalitional initial state $\hat{\mathbf{x}}_c(k)$. Likewise, $\ell_{\mathcal{C}}\big(\mathbf{x}_{\mathcal{C}}(k), \mathbf{u}_{\mathcal{C}}(k)\big)$ is a certain convex coalitional stage cost that is minimized along the prediction horizon. As a result, the sequence of the coalitional optimal control inputs over N_{p}, that is, $\mathbf{U}^*_{\mathcal{C}}(k : k+N_{\mathrm{p}}-1) = \{\mathbf{u}^*_{\mathcal{C}}(k), \ldots, \mathbf{u}^*_{\mathcal{C}}(k+N_{\mathrm{p}}-1)\}$ is obtained. Only the first control input $\mathbf{u}^*_{\mathcal{C}}(k)$ is actually applied, and the rest of elements are discarded. At the next time instant, (8) is solved again in a receding horizon fashion.

Notice that a coalition can be a singleton. In this case, each local controller $i \in \mathcal{N}$ solves an optimization problem analogous to (8) by taking $\mathcal{C} = i$. Likewise, to compute a centralized MPC scheme implemented in a distributed fashion it is enough to calculate the optimal input sequence by taking $\mathcal{C} = \mathcal{N}$ and solving (8).

III. COOPERATIVE GAMES AND THE SHAPLEY VALUE

In [21], [22], the set of links $\mathcal{E}$ was interpreted as the set of players of a cooperative game whose characteristic function assigned a value to each of the different configurations of links or *network topologies*. As commented before, in the approach proposed this paper it is assumed that the number of links is related to the agents by (3). Hence, to mitigate combinatorial explosion issues associated to LSS, the perspective of the aforementioned works is changed here, as done in [23], to working directly with a cooperative game $(\mathcal{N}, \boldsymbol{v})$ defined over the set of agents $\mathcal{N}$. To this end, a cost function $\boldsymbol{v}$ that assigns a cost to each coalition of players $\mathcal{C} \subseteq \mathcal{N}$ is defined by

$$v(\mathcal{C}, \mathbf{x}_{\mathcal{N}}) = \sum_{k=0}^{T_{\mathrm{sim}}-1} \left[\ell_{\mathcal{C}}\big(\mathbf{x}_{\mathcal{C}}(k+1), \mathbf{u}^*_{\mathcal{C}}(k)\big) + \sum_{i\notin\mathcal{C}} \ell_i\big(\mathbf{x}_i(k+1), \mathbf{u}^*_i(k)\big) \right], \tag{9}$$

with $\ell_i\big(\mathbf{x}_i(k), \mathbf{u}_i(k)\big)$ and $\ell_{\mathcal{C}}\big(\mathbf{x}_{\mathcal{C}}(k), \mathbf{u}_{\mathcal{C}}(k)\big)$ being the stage costs, which will be defined for the case study in Section V-B, and where T_{sim} is the number of simulation steps used to accumulate the closed-loop stage cost of the system over the simulation time. This function $\boldsymbol{v}$ is computed by applying at each time step the first element of the control sequence of coalition $\mathcal{C}$, i.e., $\mathbf{u}^*_{\mathcal{C}}(k)$, which is obtained by solving (8). The rest of the agents calculate their input sequences $\mathbf{u}^*_i(k)$ by solving (8), with $\mathcal{C} = i$, independently.

Remark 1. *Equation (9) is evaluated with input information from all agents, independently whether they are either in or out the coalition $\mathcal{C}$. Each coalition $\mathcal{C}$ solves its own optimization problem, which is decoupled from the rest of the network. Hence, $2^{|\mathcal{N}|}$ values are needed to fully characterize function $\boldsymbol{v}$ using the standard approach, which is not the case in this work, as it will be shown in the next subsection by considering randomized methods.*

Once the game is defined, the Shapley value [27] is considered here to get the corresponding cost that each player expects from the game. This value assigns to game $(\mathcal{N}, \boldsymbol{v})$ vector $\boldsymbol{\phi}(\mathcal{N}, \boldsymbol{v})$, defined $\forall i \in \mathcal{N}$ as

$$\phi_i(\mathcal{N}, \boldsymbol{v}) = \sum_{\mathcal{C} \subseteq \mathcal{N}, i \notin \mathcal{C}} \frac{|\mathcal{C}|!(|\mathcal{N}| - |\mathcal{C}| - 1)!}{|\mathcal{N}|!} [v(\mathcal{C} \cup \{i\}, \mathbf{x}_{\mathcal{N}}) - v(\mathcal{C}, \mathbf{x}_{\mathcal{N}})], \tag{10}$$

i.e., the marginal contribution of each agent $i \in \mathcal{N}$ is averaged for all possible coalition permutations it can be part of.

Remark 2. *Equation (10) was originally defined in the context of transferable utility (TU) games. Given that agents out of $\mathcal{C}$ work independently, an univocal $v(\mathcal{C})$ is obtained for each $\mathcal{C}$, and (9) could be interpreted as a TU-game by simply assuming a redefined game $v\prime(\mathcal{C}, \mathbf{x}_{\mathcal{N}}) = v(\mathcal{C}, \mathbf{x}_{\mathcal{N}}) - v(\emptyset, \mathbf{x}_{\mathcal{N}})$. This way, the Shapley value of this redefined game, which trivially coincides with the Shapley value of the original game (see [22]), will be considered here.*

Given that the partitioning procedure proposed in this paper will be performed by fixing/disconnecting some communication links among the different agents, a measure of the relevance of the links is required. Given a link $l = \{i, j\} \in \mathcal{E}$, it is possible to redistribute the Shapley value of the agents that are the end-points of this link, i.e., i and j, by means of the following expression [23]:

$$\xi_l(\mathcal{N}, \boldsymbol{v}) = \frac{1}{|\mathcal{E}_i|}\phi_i(\mathcal{N}, \boldsymbol{v}) + \frac{1}{|\mathcal{E}_j|}\phi_j(\mathcal{N}, \boldsymbol{v}), \tag{11}$$

with $\mathcal{E}_i$ and $\mathcal{E}_j$ being, respectively, the set of links connected to agents i and j. Notice that values $\xi_l(\mathcal{N}, \boldsymbol{v})$ satisfy *efficiency* as the original Shapley value, i.e.,

$$\sum_{l \in \mathcal{E}} \xi_l(\mathcal{N}, \boldsymbol{v}) = \sum_{i \in \mathcal{N}} \phi_i(\mathcal{N}, \boldsymbol{v}) = v(\mathcal{N}, \mathbf{x}_{\mathcal{N}}). \tag{12}$$

Note that (11) provides a way to arrange and compare the links according to their relevance from a control-performance perspective, which can be interpreted as a *LinkRank*, in the line of [32]. This way, the lower this value is, the more useful the link becomes. This is consistent with the Shapley value, which associates useful players to lower values.

Remark 3. *Equation (11) provides information from all links $l \in \mathcal{E}$ by means of the Shapley value of agent-based game (9), which is obtained by evaluating the corresponding $2^{|\mathcal{N}|}$ coalitions. This fact mitigates the combinatorial explosion of the method proposed in [21], which depends on a link-game that requires to evaluate $2^{|\mathcal{E}|}$ coalitions per game, with $|\mathcal{E}| = f(|\mathcal{N}|^2)$ as shown in (3).*

A. ESTIMATION OF THE SHAPLEY VALUE

In networks with a large number of agents it is not computationally feasible to compute (9) for every coalition. This issue can be solved by using randomized methods such as [28], [29]. In particular, the method in [28] is used here to provide an estimation of the Shapley value of each agent calculated in polynomial time provided that (9) can also be calculated in polynomial time, which will be shown in Section V-B. To this end, the following alternative definition of the Shapley value, expressed in terms of all possible orderings of players in $\mathcal{N}$ coming into coalition, i.e., $|\mathcal{N}|!$, is used in the sampling method given in [28]:

$$\phi_i(\mathcal{N}, \boldsymbol{v}) = \frac{1}{|\mathcal{N}|!} \sum_{\pi \in \Pi(\mathcal{N})} m_i^{\pi}(\mathcal{N}, \boldsymbol{v}), \quad \forall i \in \mathcal{N}, \tag{13}$$

where the orderings are assumed to be equiprobable, with $\Pi(\mathcal{N})$ being the collection of all permutations π, and where

$$\begin{aligned} m_i^{\pi}(\mathcal{N}, \boldsymbol{v}) \;=\;& v\left(\{j \in \mathcal{N} : \pi(j) \leq \pi(i)\}\right) \\ -\;& v(\{j \in \mathcal{N} : \pi(j) < \pi(i)\}), \end{aligned} \tag{14}$$

is the marginal contribution of player i to the players that are ranked before it in permutation π.

The basics of the method in [28] consist in choosing a given number of random orderings from set $\Pi(\mathcal{N})$ for estimating the Shapley value of each player. To this end, a set $\mathcal{Q}$ that contains a sample of q different permutations π, which are taken with replacement and with equal probability from set $\Pi(\mathcal{N})$, is considered. This way, an estimation of the Shapley value is given by the average of the marginal contributions over set $\mathcal{Q}$, i.e.,

$$\widetilde{\phi}_i(\mathcal{N}, \boldsymbol{v}) = \frac{1}{q} \sum_{\pi \in \mathcal{Q}} m_i^{\pi}(\mathcal{N}, \boldsymbol{v}), \quad \forall i \in \mathcal{N}. \tag{15}$$

Equation (15) provides an approximation of the Shapley value with desirable properties. In particular, as the Shapley value, the estimator satisfies *efficiency*. Moreover, following the central limit theorem, it holds [28] that the estimator is a normal distribution with the following mean value and standard deviation:

$$\widetilde{\phi}_i(\mathcal{N}, \boldsymbol{v}) \sim N\left(\phi_i, \frac{\sigma_{\phi_i}^2}{q}\right), \tag{16}$$

with

$$\sigma_{\phi_i}^2 = \frac{1}{|\mathcal{N}|!} \sum_{\pi \in \Pi(\mathcal{N})} \left(m_i^{\pi}(\mathcal{N}, \boldsymbol{v}) - \phi_i(\mathcal{N}, \boldsymbol{v})\right)^2, \quad \forall i \in \mathcal{N}. \tag{17}$$

Consequently, if the number of permutations q is chosen satisfying the following condition, $\forall i \in \mathcal{N}$ [28]:

$$q \geq \frac{Z_{\lambda/2}^2 \sigma_{\phi_i}^2}{\varepsilon^2}, \tag{18}$$

the estimation error is is guaranteed to be bounded by [28]

$$P\big(|\widetilde{\phi}_i(\mathcal{N}, \boldsymbol{v}) - \phi_i(\mathcal{N}, \boldsymbol{v})| \leq \varepsilon\big) \geq 1 - \lambda, \tag{19}$$

with ε being the approximation error, $Z \sim N(0, 1)$, and where $Z_{\lambda/2}^2$ is the value such that $P(Z \geq Z_{\lambda/2}^2) = \lambda/2$, with $0 \leq \lambda \leq 1$. Finally, note that by taking $\sigma_{\phi_i} = \alpha\varepsilon$, i.e., proportionally to the desired error, condition (18) is reduced to

$$q \geq \alpha Z_{\lambda/2}^2. \tag{20}$$

IV. SHAPLEY-VALUE-BASED PARTITIONING ALGORITHM

The main objective of the algorithm presented in this paper is to find which agents should cooperate to improve the overall system performance. Notice that, some preliminar steps are needed before it can be launched. In the first place, there is a pre-partitioning stage in which a set of atomic agents needs to be defined by assigning states, actions and constraints to minimum size entities that could work in a fully decentralized fashion, i.e., without the need for communication with other parts of the system. In this work, as it will be shown in Section V-A, these agents stem from the constraints imposed on the system by the node equations.

Once the agents are defined, a communication link between each pair of agents is considered, with the full number of links given by (3). Then, it is needed to classify these links according to their relevance in terms of their impact on the overall system performance. To this end, a measure index related to the Shapley value is associated with each link as follows:

Measure Indices Procedure

Let $(\mathcal{N},\mathcal{E})$ be a network that describes a set $\mathcal{N}$ of agents connected by links $l \in \mathcal{E}$. Consider also a measured initial state $\hat{\mathbf{x}}_{\mathcal{N}}(k)$ and a forecast of the expected disturbances $\hat{\mathbf{W}}_{\mathcal{N}}(k : k + N_{\rm p} - 1)$. Then,

a) Calculate a size q that guarantees, following (18), that the estimation error is under desired limits.

b) Compute $\widetilde{\phi}_i(\mathcal{N},\boldsymbol{v}), \forall i \in \mathcal{N}$, by using (15). For each coalition $\mathcal{C} \in \mathcal{N}_{\mathcal{C}}$, the optimal input sequence over $N_{\rm p}$ is obtained by solving (8) for $\mathcal{C}$ and also for the agents out of $\mathcal{C}$. Only the first control input is applied, and the rest of elements are discarded. At the next time step these optimization problems are solved in a receding horizon fashion. This process is performed during $T_{\rm sim}$ time instants and the cumulated cost of this closed-loop simulation is used to build $v(\mathcal{C},\mathbf{x}_{\mathcal{N}})$ by means of (9).

c) Redistribute the obtained Shapley value among the links by indices $\widetilde{\xi}_l(\mathcal{N},\boldsymbol{v})$, calculated by using (11).

Therefore, index $\widetilde{\xi}_l(\mathcal{N},\boldsymbol{v})$ measures the impact of link l in the control network. Based on these indices, it is possible to rank the links, which in turns allows to obtain the following subsets:

- Set $\mathcal{E}_{\rm c} \subseteq \mathcal{E}$: it includes the links that are inexpensive enough in control terms to always be fixed. This way, agents connected by links belonging to $\mathcal{E}_{\rm c}$ cooperate together and will be merged in a single agent, which corresponds to any of the communication components in $\mathcal{N}_{\mathcal{C}} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{n_{\rm c}}\}$.
- Set $\mathcal{E}_{\rm e} \subseteq \mathcal{E}$: it comprises the links that are too costly for the system in terms of control performance, and therefore will be permanently disconnected.

Basically, the partitioning algorithm in this section is introduced with the aim of determining sets $\mathcal{E}_{\rm c}$ and $\mathcal{E}_{\rm e}$. Both sets are initially assumed to be empty, and links will be gradually included in those sets if they satisfy certain conditions. This way, links in $\mathcal{E}_{\rm c}(s)$ define the communication components in $\mathcal{N}_{\mathcal{C}}(s)$ in a given iteration $s \in \mathbb{N}^+$. Some concepts that will be needed to perform the partitioning are introduced next:

- K: symbolizes the mean span among indices $\widetilde{\xi}_l(\mathcal{N},\boldsymbol{v})$ and it is defined by
$$K = \frac{\widetilde{\xi}_{l_{\max}}(\mathcal{N},\boldsymbol{v}) - \widetilde{\xi}_{l_{\min}}(\mathcal{N},\boldsymbol{v})}{|\mathcal{E}|}, \tag{21}$$
where $l_{\max}$ and $l_{\min}$ denote, respectively, the links with maximum and minimum measure indices $\widetilde{\xi}_l(\mathcal{N},\boldsymbol{v})$.
- $\mathcal{C}_i(s)$: denotes the component in $\mathcal{N}_{\mathcal{C}}(s)$ where agent i belongs to.
- $\mathcal{E}_i(s)$: denotes the set of remaining links in $\mathcal{E}\backslash\mathcal{E}_{\rm c}(s)$ connected to agent i.

Notice that the sizes of sets $\mathcal{C}_i(s)$ and $\mathcal{E}_i(s)$ are inversely related. Both sets will be of interest in the Shapley-value-based partitioning algorithm, which is presented below.

$$\text{if } \widetilde{\xi}^{(s)}_{l_{\rm c}^*}(\mathcal{N},\boldsymbol{v}) + \Gamma < \mathscr{L}_{\rm c} \textbf{ AND } \mathcal{C}_{i^*}(s) \neq \mathcal{C}_{j^*}(s) \Longrightarrow \begin{cases} \mathcal{E}_{\rm c}(s+1) = \mathcal{E}_{\rm c}(s) \cup l_{\rm c}^*, \\ \widetilde{\xi}^{(s+1)}_l(\mathcal{N},\boldsymbol{v}) = \begin{cases} \widetilde{\xi}^{(s)}_l(\mathcal{N},\boldsymbol{v}), & l \notin \{\mathcal{E}_{i^*}(s+1) \cup \mathcal{E}_{j^*}(s+1)\}, \\ \widetilde{\xi}^{(s)}_l(\mathcal{N},\boldsymbol{v}) + \frac{\rho K}{|\mathcal{E}_{i^*}(s+1)|}, & l \in \mathcal{E}_{i^*}(s+1), \\ \widetilde{\xi}^{(s)}_l(\mathcal{N},\boldsymbol{v}) + \frac{\rho K}{|\mathcal{E}_{j^*}(s+1)|}, & l \in \mathcal{E}_{j^*}(s+1), \end{cases} \end{cases}$$
$$\text{else if } \mathcal{C}_{i^*}(s) = \mathcal{C}_{j^*}(s) \Longrightarrow \begin{cases} \mathcal{E}_{\rm c}(s+1) = \mathcal{E}_{\rm c}(s) \cup l_{\rm c}^*, \\ \widetilde{\xi}^{(s+1)}_l(\mathcal{N},\boldsymbol{v}) = \widetilde{\xi}^{(s)}_l(\mathcal{N},\boldsymbol{v}), \end{cases}$$
$$\text{otherwise} \Longrightarrow \begin{cases} \mathcal{E}_{\rm c}(s+1) = \mathcal{E}_{\rm c}(s), \\ \widetilde{\xi}^{(s+1)}_l(\mathcal{N},\boldsymbol{v}) = \widetilde{\xi}^{(s)}_l(\mathcal{N},\boldsymbol{v}), \end{cases} \tag{23}$$

Partitioning Algorithm 1

Let $\widetilde{\xi}_l(\mathcal{N}, \boldsymbol{v})$ be the indices related to each link $l \in \mathcal{E}$. Let $\mathscr{L}_{\rm c}, \mathscr{L}_{\rm e} \in \mathbb{R}$ be given thresholds, verifying $\mathscr{L}_{\rm c} \leq \mathscr{L}_{\rm e}$. Finally, let s be a counter variable.

I) Computation of set $\mathcal{E}_{\rm c}$

Let $\gamma, \rho \in \mathbb{R}^+ \backslash \{0\}$, $\nu \geq 1$ be scalar parameters. Let also K be the mean span given by (21). Starting with $s = 0$, and assuming $\widetilde{\xi}_l^{(0)}(\mathcal{N}, \boldsymbol{v}) = \widetilde{\xi}_l(\mathcal{N}, \boldsymbol{v})$ and $\mathcal{E}_{\rm c}(0) = \emptyset$, do

1) Obtain the link with minimum measure index from the remaining links that are not yet included in set $\mathcal{E}_{\rm c}(s)$, i.e.,
$$l_{\rm c}^* = \arg\min_l \widetilde{\xi}_l^{(s)}(\mathcal{N}, \boldsymbol{v}), \quad l \in \mathcal{E} \backslash \mathcal{E}_{\rm c}(s). \tag{22}$$
2) Validate link $l_{\rm c}^* = \{i^*, j^*\}$ as a suitable candidate to be added to $\mathcal{E}_{\rm c}(s)$, and update this set and the links neighbors of $l_{\rm c}^*$ consequently, doing, $\forall l \in \mathcal{E} \backslash \mathcal{E}_{\rm c}(s+1)$, the procedure defined by (23), with Γ being calculated as
$$\Gamma = \gamma K \left(|\mathcal{C}_{i^*}(s)| + |\mathcal{C}_{j^*}(s)|\right)^{\nu}. \tag{24}$$
3) Make $s = s + 1$ and go to Step 1, while $\widetilde{\xi}_{l_{\rm c}^*}^{(s)}(\mathcal{N}, \boldsymbol{v}) < \mathscr{L}_{\rm c}$. Otherwise, the procedure ends and set $\mathcal{E}_{\rm c}$ is fully determined, i.e., $\mathcal{E}_{\rm c} = \mathcal{E}_{\rm c}(s)$.

II) Computation of set $\mathcal{E}_{\rm e}$

Starting again with $s = 0$, and assuming $\widetilde{\xi}_l^{(0)}(\mathcal{N}, \boldsymbol{v}) = \widetilde{\xi}_l(\mathcal{N}, \boldsymbol{v})$, $\mathcal{E}_{\rm e}(0) = \emptyset$, do

4) Obtain the link with maximum measure index from the remaining links, i.e.,
$$l_{\rm e}^* = \arg\max_l \widetilde{\xi}_l^{(s)}(\mathcal{N}, \boldsymbol{v}), \quad l \in \mathcal{E} \backslash (\mathcal{E}_{\rm c} \cup \mathcal{E}_{\rm e}(s)). \tag{25}$$
5) Validate link $l_{\rm e}^* = \{i^*, j^*\}$ as a suitable candidate for set $\mathcal{E}_{\rm e}(s)$, and update this set consequently, doing
$$\begin{array}{ll} \mathcal{E}_{\rm e}(s+1) = \mathcal{E}_{\rm e}(s) \cup l_{\rm e}^*, & \text{if } \widetilde{\xi}_{l_{\rm e}^*}^{(s)}(\mathcal{N}, \boldsymbol{v}) > \mathscr{L}_{\rm e}, \\ \mathcal{E}_{\rm e}(s+1) = \mathcal{E}_{\rm e}(s), & \text{otherwise.} \end{array} \tag{26}$$
6) Make $s = s + 1$ and go to Step 4, while $\widetilde{\xi}_{l_{\rm e}^*}^{(s)}(\mathcal{N}, \boldsymbol{v}) > \mathscr{L}_{\rm e}$. Otherwise, the procedure ends and set $\mathcal{E}_{\rm e}$ is fully determined, i.e., $\mathcal{E}_{\rm e} = \mathcal{E}_{\rm e}(s)$.

Notice that the inclusion of a new link $l_{\rm c}^*$ in $\mathcal{E}_{\rm c}(s)$ depends on the size of the components at s that this link will connect. More specifically, Γ penalizes a new link candidate to $\mathcal{E}_{\rm c}(s)$ before deciding whether it should be included in that set, in case that this link would connect two different components in $\mathcal{N}_{\mathcal{C}}(s)$, and proportionally to their cardinality. Additionally, in case that $l_{\rm c}^*$ is accepted, the proposed algorithm penalizes its neighboring links that remain in set $\mathcal{E} \backslash \mathcal{E}_{\rm c}(s+1)$, by a term that is larger as less neighbors of $l_{\rm c}^*$ remain in that set, which in turns implies that $l_{\rm c}^*$ is more congested in $\mathcal{E}_{\rm c}(s+1)$. These mechanisms induce size constraints on the components, which avoid *inefficient* partitionings that could lead to an almost centralized scheme. This way, the proposed parameters γ, ν and ρ could be adjusted to obtain some properties of interest, e.g., to impose a maximum cardinality for any communication component.

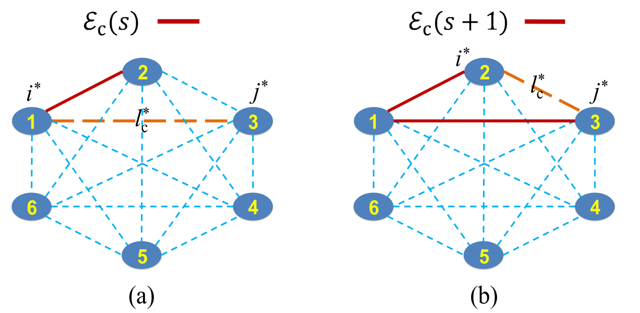

FIGURE 1. Two iteration steps in the process of obtaining set $\mathcal{E}_{\rm c}$

Example 1. *Take the network in Fig. 1, with six agents connected by 15 links. In a given iteration s (see Fig. 1a), it is obtained*
$$\mathcal{E}_{\rm c}(s) = \{\{1,2\}\}, \quad \mathcal{N}_{\mathcal{C}}(s) = \{\{1,2\},\{3\},\{4\},\{5\},\{6\}\}, \tag{27}$$
Then, consider that link $l_{\rm c}^* = \{i^*, j^*\} = \{1,3\}$ is proposed to be fixed. The sets related to this link in iteration s are described by
$$\begin{array}{ll} \mathcal{C}_1(s) = \{1,2\}, & \mathcal{E}_1(s) = \{\{1,3\},\{1,4\},\{1,5\},\{1,6\}\}, \\ \mathcal{C}_3(s) = \{3\}, & \mathcal{E}_3(s) = \{\{1,3\},\{2,3\},\{3,4\},\{3,5\},\{3,6\}\}. \end{array} \tag{28}$$
Assume at this point that $\widetilde{\xi}_{l_{\rm c}^*}^{(s)}(\mathcal{N}, \boldsymbol{v}) + \Gamma < \mathscr{L}_{\rm c}$. This way, given that $\mathcal{C}_1(s) \neq \mathcal{C}_3(s)$, the first statement in (23) is fulfilled, hence link $\{1,3\}$ is included in set $\mathcal{E}_{\rm c}(s+1)$ (see Fig. 1b), obtaining
$$\begin{array}{c} \mathcal{E}_{\rm c}(s+1) = \{\{1,2\},\{1,3\}\}, \\ \mathcal{N}_{\mathcal{C}}(s+1) = \{\{1,2,3\},\{4\},\{5\},\{6\}\}, \\ \mathcal{C}_1(s+1) = \mathcal{C}_3(s+1) = \{1,2,3\}, \\ \mathcal{E}_1(s+1) = \{\{1,4\},\{1,5\},\{1,6\}\}, \\ \mathcal{E}_3(s+1) = \{\{2,3\},\{3,4\},\{3,5\},\{3,6\}\}. \end{array} \tag{29}$$
Therefore, $\forall l \in \mathcal{E} \backslash \mathcal{E}_{\rm c}(s+1)$, all measure indices are updated if l belongs either to $\mathcal{E}_1(s+1)$ or $\mathcal{E}_3(s+1)$, by increasing in an inversely proportional way to the cardinality of those sets
$$\begin{array}{ll} \widetilde{\xi}_l^{(s+1)}(\mathcal{N}, \boldsymbol{v}) = \widetilde{\xi}_l^{(s)}(\mathcal{N}, \boldsymbol{v}) + \frac{\rho K}{3}, & l \in \mathcal{E}_1(s+1), \\ \widetilde{\xi}_l^{(s+1)}(\mathcal{N}, \boldsymbol{v}) = \widetilde{\xi}_l^{(s)}(\mathcal{N}, \boldsymbol{v}) + \frac{\rho K}{4}, & l \in \mathcal{E}_3(s+1). \end{array} \tag{30}$$
Finally, consider that $l_{\rm c}^* = \{i^*, j^*\} = \{2,3\}$ is also proposed to be fixed. In that case, given that $\mathcal{C}_2(s+1) = \mathcal{C}_3(s+1) = \{1,2,3\}$, the second statement in (25) is satisfied, and the link will be included in set $\mathcal{E}_{\rm c}(s+2)$. Nevertheless, no updates on the measure indices of links $l \in \mathcal{E} \backslash \mathcal{E}_{\rm c}(s+2)$ would be considered.

Once set $\mathcal{E}_\mathrm{c}$ is fully determined, a new single agent corresponding to each component in $\mathcal{N}_\mathcal{C}$ is established, independently of whether the agents inside this component are directly or indirectly connected. Therefore, the system is reduced to $|\mathcal{N}_\mathcal{C}|$ new agents, which may communicate or not, depending on the needs of the control scheme. The possible links among the agents are denoted by $\mathcal{E}_\mathcal{C} = \mathcal{N}_\mathcal{C} \times \mathcal{N}_\mathcal{C}$, which verifies $|\mathcal{E}_\mathcal{C}| = \frac{|\mathcal{N}_\mathcal{C}|(|\mathcal{N}_\mathcal{C}|-1)}{2}$, according to (3). That is, links $l_\mathcal{C} \in \mathcal{E}_\mathcal{C}$ are defined as

$$l_\mathcal{C} = \{\mathcal{C}_\mathrm{a}, \mathcal{C}_\mathrm{b}\}, \quad \forall \mathcal{C}_\mathrm{a}, \mathcal{C}_\mathrm{b} \in \mathcal{N}_\mathcal{C}. \tag{31}$$

Finally, the information provided by $\mathcal{E}_\mathrm{e}$ is used to determine links $l_\mathcal{C}$ that should be permanently disconnected. Given that links in $\mathcal{E}_\mathrm{e}$ are referred to agents instead of components, it is necessary to obtain a new set, say $\mathcal{E}_{\mathrm{e}_\mathcal{C}}$, referred to components. In this work, it is considered that two components should not have a direct cooperation if all links that interconnect the agents inside both components belong to subset $\mathcal{E}_\mathrm{e}$, i.e.,

$$\text{If } l = \{i,j\} \in \mathcal{E}_\mathrm{e}, \forall i \in \check{\mathcal{C}}_\mathrm{a}, \forall j \in \check{\mathcal{C}}_\mathrm{b} \longrightarrow \check{l}_\mathcal{C} = \{\check{\mathcal{C}}_\mathrm{a}, \check{\mathcal{C}}_\mathrm{b}\} \in \mathcal{E}_{\mathrm{e}_\mathcal{C}}. \tag{32}$$

Summing up, the configuration of the system will be described after performing the partitioning by the following network:

$$(\mathcal{N}_\mathcal{C}, \mathcal{E}_\mathcal{C} \backslash \mathcal{E}_{\mathrm{e}_\mathcal{C}}), \tag{33}$$

where the links in $\mathcal{E}_\mathcal{C} \backslash \mathcal{E}_{\mathrm{e}_\mathcal{C}}$ may be dynamically enabled or disabled at each time instant by means of a coalitional control approach [22], [24].

Remark 4. *The proposed Shapley-value-based algorithm represents a heuristic methodology for the system partitioning that avoids an exhaustive exploration of every coalition involved in the control scheme. Note that this methodology is independent from the model dynamics or the game choice, i.e., nonlinear systems or alternative definitions for the game different to (9) are possible. For example, it might be considered theoretical aspects such as stability or robustness [2], [33], which are out of the scope here since this work only focus on the partitioning methodology. Likewise, for the sake of clarity, the class of linear systems described by (1), which is widely studied in the literature, is assumed here.*

Remark 5. *The way the agents inside a component are actually connected once the partitioning is performed is beyond of the scope of this work. This issue could be dealt with, e.g., by using spanning tree algorithms [34], [35], in order to find the minimum set of links that is necessary to connect all agents belonging to a given component. In any case, note that the partitioning approach reduces the communication costs of the original centralized scheme, given that the agents inside a component after performing the partitioning are only required to communicate to their neighbors.*

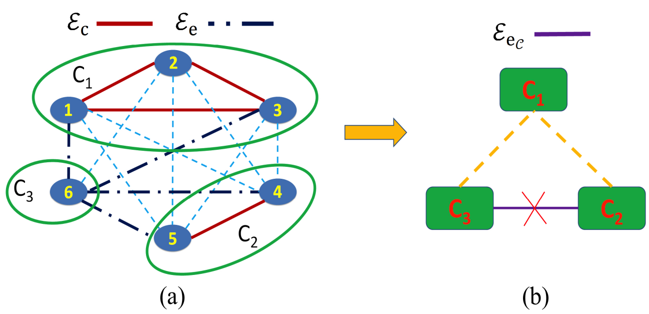

FIGURE 2. Establishing the components and their removed links by $\mathcal{E}_\mathrm{c}$ and $\mathcal{E}_\mathrm{e}$

Example 2. *Consider again the network presented in Example 1, with six agents and 15 links. Assume that after applying the partitioning algorithm, sets $\mathcal{E}_\mathrm{c}$ and $\mathcal{E}_\mathrm{e}$ are given by (see Fig. 2a):*

$$\begin{aligned} \mathcal{E}_\mathrm{c} &= \{\{1,2\},\{1,3\},\{2,3\},\{4,5\}\}, \\ \mathcal{E}_\mathrm{e} &= \{\{1,6\},\{3,6\},\{4,6\},\{5,6\}\}. \end{aligned} \tag{34}$$

The links in $\mathcal{E}_\mathrm{c}$ define the following components:

$$\mathcal{C}_1 = \{1,2,3\}, \;\; \mathcal{C}_2 = \{4,5\}, \;\; \mathcal{C}_3 = \{6\}, \tag{35}$$

which are also assumed to be connected by links $\{\mathcal{C}_1,\mathcal{C}_2\}$, $\{\mathcal{C}_1,\mathcal{C}_3\}$ and $\{\mathcal{C}_2,\mathcal{C}_3\}$. From these three links, only link $\{\mathcal{C}_2,\mathcal{C}_3\}$ verifies the criterion given in (32), i.e., all links that connect agents in components $\mathcal{C}_2$ and $\mathcal{C}_3$ are included in set $\mathcal{E}_\mathrm{e}$. Hence, this link should be permanently disconnected. Therefore, the final configuration of the network is shown in Fig. 2b and it is given by

$$(\mathcal{N}_\mathcal{C}, \mathcal{E}_\mathcal{C} \backslash \mathcal{E}_{\mathrm{e}_\mathcal{C}}) = (\{\mathcal{C}_1,\mathcal{C}_2,\mathcal{C}_3\}, \{\{\mathcal{C}_1,\mathcal{C}_2\},\{\mathcal{C}_1,\mathcal{C}_3\}\}). \tag{36}$$

A. SENSITIVITY ANALYSIS OF THE CONTROL PERFORMANCE: A PARTITIONING ALGORITHM FINE TUNING

The rationale behind the partitioning procedure is to provide a reasonable trade-off between control and communication costs. Therefore, the proposed Shapley-value-based approach, denoted from now on by SVBA, provides us with a suboptimal solution $\mathcal{N}_\mathcal{C}$ for the partitioning of an LSS, where the optimal solution corresponds with the centralized case (only one component) when communication costs are not considered. Once $\mathcal{N}_\mathcal{C}$ is established, the closed-loop system performance can be related to the cumulated cost $J_\mathrm{cum}(\mathcal{N}_\mathcal{C})$, which is obtained by computing each component $\mathcal{C} \in \mathcal{N}_\mathcal{C}$ by using (9). Then, in order to give an insight of the SVBA fitness, it would be interesting to compare the cumulated cost of partition $\mathcal{N}_\mathcal{C}$ with the rest of possible partitioning approaches. Nevertheless, note that the number of ways to partition a set of $|\mathcal{N}|$ agents into nonempty components is given by the Bell number [36]

$$B_{|\mathcal{N}|} = \sum_{s=0}^{|\mathcal{N}|} \left(\frac{1}{s!} \sum_{j=0}^{s} (-1)^{s-j} \binom{s}{j} j^{|\mathcal{N}|} \right), \tag{37}$$

which becomes computationally infeasible for LSS so as comparing $\mathcal{N}_\mathcal{C}$ with this full set. To solve this issue, here it is considered the subset composed of all partitions $\mathcal{N}_\mathcal{C}^{\text{swi}}$, which differ from $\mathcal{N}_\mathcal{C}$ in the fact that *only* one agent $i \in \mathcal{N}$ switches components. This set will be denoted by $\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}$ and its cardinality is given by

$$|\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}| = |\mathcal{N}||\mathcal{N}_\mathcal{C}| - |\mathcal{N}_\mathcal{C}^{|\mathcal{C}|=1}|\left(\frac{|\mathcal{N}_\mathcal{C}^{|\mathcal{C}|=1}|+1}{2}\right), \tag{38}$$

where $|\mathcal{N}_\mathcal{C}^{|\mathcal{C}|=1}|$ is the number of singletons in $\mathcal{N}_\mathcal{C}$. Note that a new component could be formed assuming the agent that switches components is not already a singleton in $\mathcal{N}_\mathcal{C}$. Likewise, redundant switches between any two singletons are also discarded from set $\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}$.

Once set $\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}$ is introduced, some related indices will be obtained and compared with those of $\mathcal{N}_\mathcal{C}$, which can be interpreted as a sensitivity analysis of the SVBA. In particular, the minimum, maximum, and mean cumulated costs of partitions $\mathcal{N}_\mathcal{C}^{\text{swi}} \in \Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}$ will be of interest. Likewise, a parameter that computes the portion of partitions $\mathcal{N}_\mathcal{C}^{\text{swi}}$ that are improved by $\mathcal{N}_\mathcal{C}$ will also be considered and symbolized by η_p, with p being a certain uncertainty limit, i.e., to be better than $\mathcal{N}_\mathcal{C}$ it is required to outperform its cost beyond a certain threshold.

Notice also that the partition with minimum cumulated cost from $\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}$, say $\mathcal{N}_\mathcal{C}^{(1)}$, provides a suboptimal solution that improves $\mathcal{N}_\mathcal{C}$. Consequently, it is possible to optimize recursively the SVBA in a greedy fashion by using the minimum-cost solutions $\mathcal{N}_\mathcal{C}^{(r)}$ from the successive sets $\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}^{(r-1)}$, whose elements in turns admit that only one agent switches components from partition $\mathcal{N}_\mathcal{C}^{(r-1)}$, with $r \in \mathbb{N}^+, r > 1$, i.e.,

$$\begin{aligned} \mathcal{N}_\mathcal{C}^{(r)} &= \arg\min_{\mathcal{N}_\mathcal{C}^{\text{swi}}} J_{\text{cum}}(\mathcal{N}_\mathcal{C}^{\text{swi}}) \\ &\text{s.t. } \mathcal{N}_\mathcal{C}^{\text{swi}} \in \Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}^{(r-1)}, \end{aligned} \tag{39}$$

until reaching any pre-established stopping criterion, e.g., a maximum number of iterations $r_{\max}$ or a minimum performance improvement between iterations. This optimization represents a *fine tuning* of the SVBA, denoted here by SVBA-FT. Note that, as considered in the SVBA, some additional constraints should be included by the control designer in the SVBA-FT to balance the size of the components. Likewise, note that sets $\mathcal{E}_\text{c}$, $\mathcal{E}_\text{e}$ and $\mathcal{E}_{\text{ec}}$ could be modified as a consequence of the fine tuning procedure. In any case, these changes improve the performance of the optimized solution, symbolized by $\mathcal{N}_\mathcal{C}^{\text{opt}}$.

Remark 6. *Given how $\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}$ is built, the impact on communication burden between two consecutive optimization steps is negligible. Nevertheless, when a high number of steps is performed, the aforementioned size constraints and also a stopping criterion are necessary to avoid inefficient centralized partitionings.*

TABLE 1. Set $\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}$ related to $\mathcal{N}_\mathcal{C}$ described by (35)

1	$\mathcal{C}_1=\{2,3\}$	$\mathcal{C}_2=\{1,4,5\}$	$\mathcal{C}_3=\{6\}$	
2	$\mathcal{C}_1=\{2,3\}$	$\mathcal{C}_2=\{4,5\}$	$\mathcal{C}_3=\{1,6\}$	
3	$\mathcal{C}_1=\{2,3\}$	$\mathcal{C}_2=\{4,5\}$	$\mathcal{C}_3=\{6\}$	$\mathcal{C}_4=\{1\}$
4	$\mathcal{C}_1=\{1,3\}$	$\mathcal{C}_2=\{2,4,5\}$	$\mathcal{C}_3=\{6\}$	
5	$\mathcal{C}_1=\{1,3\}$	$\mathcal{C}_2=\{4,5\}$	$\mathcal{C}_3=\{2,6\}$	
6	$\mathcal{C}_1=\{1,3\}$	$\mathcal{C}_2=\{4,5\}$	$\mathcal{C}_3=\{6\}$	$\mathcal{C}_4=\{2\}$
7	$\mathcal{C}_1=\{1,2\}$	$\mathcal{C}_2=\{3,4,5\}$	$\mathcal{C}_3=\{6\}$	
8	$\mathcal{C}_1=\{1,2\}$	$\mathcal{C}_2=\{4,5\}$	$\mathcal{C}_3=\{3,6\}$	
9	$\mathcal{C}_1=\{1,2\}$	$\mathcal{C}_2=\{4,5\}$	$\mathcal{C}_3=\{6\}$	$\mathcal{C}_4=\{3\}$
10	$\mathcal{C}_1=\{1,2,3,4\}$	$\mathcal{C}_2=\{5\}$	$\mathcal{C}_3=\{6\}$	
11	$\mathcal{C}_1=\{1,2,3\}$	$\mathcal{C}_2=\{5\}$	$\mathcal{C}_3=\{4,6\}$	
12	$\mathcal{C}_1=\{1,2,3\}$	$\mathcal{C}_2=\{5\}$	$\mathcal{C}_3=\{6\}$	$\mathcal{C}_4=\{4\}$
13	$\mathcal{C}_1=\{1,2,3,5\}$	$\mathcal{C}_2=\{4\}$	$\mathcal{C}_3=\{6\}$	
14	$\mathcal{C}_1=\{1,2,3\}$	$\mathcal{C}_2=\{4\}$	$\mathcal{C}_3=\{5,6\}$	
15	$\mathcal{C}_1=\{1,2,3\}$	$\mathcal{C}_2=\{4\}$	$\mathcal{C}_3=\{6\}$	$\mathcal{C}_4=\{5\}$
16	$\mathcal{C}_1=\{1,2,3,6\}$	$\mathcal{C}_2=\{4,5\}$		
17	$\mathcal{C}_1=\{1,2,3\}$	$\mathcal{C}_2=\{4,5,6\}$		

Example 3. *Let the solution of the SVBA described by (35). Following (37), the full set of different solutions for six agents is given by $B_6 = 203$, whereas the cardinality of $\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}$ is reduced to 17, according to (38). All partitions included in that set are detailed in Table 1, where the switching agent for each case with respect to $\mathcal{N}_\mathcal{C}$ is represented in blue color. Note that the cases of any agent in $\mathcal{C}_1$ or $\mathcal{C}_2$ switching to a new singleton $\mathcal{C}_4$ are taking into account. Consider for instance that $\mathcal{N}_\mathcal{C}^{\text{opt}}$, i.e., the solution after performing the fine tuning, is given by Partition 14 in Table 1 as*

$$\mathcal{C}_1^{\text{opt}} = \{1,2,3\}, \quad \mathcal{C}_2^{\text{opt}} = \{4\}, \quad \mathcal{C}_3^{\text{opt}} = \{5,6\}. \tag{40}$$

Then, note that link $\{5,6\}$ should be removed from set $\mathcal{E}_\text{e}$. Consequently, set $\mathcal{E}_{\text{ec}}^{\text{opt}}$ and the final network configuration for the SVBA-FT would be respectively described by

$$\mathcal{E}_{\text{ec}}^{\text{opt}} = \emptyset, \tag{41a}$$

$$(\mathcal{N}_\mathcal{C}^{\text{opt}}, \mathcal{E}_\mathcal{C}^{\text{opt}} \backslash \mathcal{E}_{\text{ec}}^{\text{opt}}) = \left(\left\{\mathcal{C}_1^{\text{opt}}, \mathcal{C}_2^{\text{opt}}, \mathcal{C}_3^{\text{opt}}\right\}, \mathcal{E}_\mathcal{C}^{\text{opt}}\right), \tag{41b}$$

with $\mathcal{E}_\mathcal{C}^{\text{opt}} = \mathcal{N}_\mathcal{C}^{\text{opt}} \times \mathcal{N}_\mathcal{C}^{\text{opt}}$.

V. CASE STUDY

The proposed partitioning scheme has been implemented in the Barcelona DWN, which is managed by Aguas de Barcelona, S.A. (AGBAR). The Barcelona DWN distributes the water supplied by the Ter and Llobregat rivers, which are regulated at their head by dams with an overall capacity of 600 hm^3, to the whole Barcelona metropolitan area. Besides the rivers, some additional underground wells also contribute to an overall flow of around 7 m^3/s, which becomes potable by using four drinking water treatment plants. Given the limits in the water flow provided by each source, there exist different water prices depending on water treatments and legal extraction canons.

A. COALITIONAL CONTROL MODEL

Control-oriented schemes for DWNs have been widely analyzed in the literature [37], [38]. In particular, the control approach of the full Barcelona DWN discussed in [13], [14]

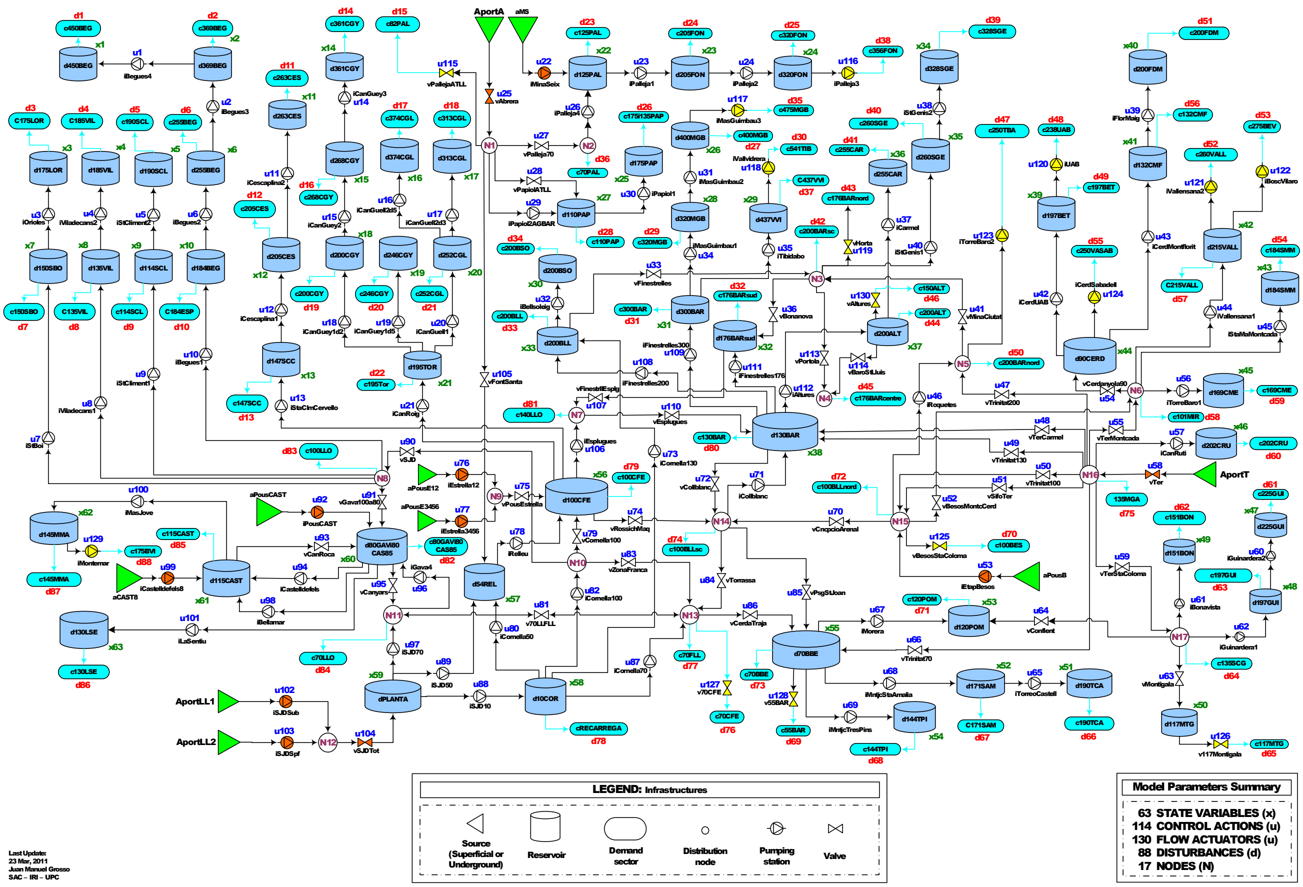

FIGURE 3. Entire model of the Barcelona DWN [13], [14]

is considered in this paper and depicted in Fig. 3. This model contains a total amount of 63 tanks, 114 actuators – divided into 75 pumps and 39 valves – and 88 sectors of consume that represent the external disturbances. A graph that summarizes the physical connections among the storage tanks and the junction nodes is provided in Fig. 4.

In the approach proposed in this paper, as commented before, an initial pre-partitioning into agents is performed due to the node equations that appear in the system (see Fig. 3), e.g.,

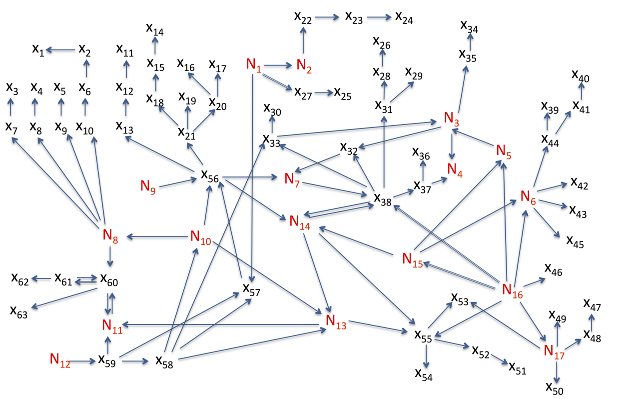

FIGURE 4. Graph representing the full Barcelona DWN, where the tanks are denoted by x, the junction nodes by N, and with the arrows representing the directions of the flows.

$$\begin{aligned} u(25) - u(27) - u(28) - u(29) - u(105) - d(15) &= 0, \\ u(27) - u(26) - d(36) &= 0, \end{aligned} \tag{42}$$

physically connect flows $u(25)$, $u(26)$, $u(27)$, $u(28)$, $u(29)$ and $u(105)$. Hence, the values of these flows must be simultaneously determined. For this reason, they are assigned to the same agent. As a consequence, the only coupling among the subsystems is due to the inputs effect in the dynamical model. Therefore, in the case study, $\mathbf{A}_{ij} = 0$ in (1), and equivalently, $\mathbf{A}_{\mathcal{C}j} = 0$ in (4). Considering this approach, 43 agents have been obtained, where the criterion of considering outflows as disturbances has been assumed, i.e., agents control their inflows. The explicit definition of each agent is included in Appendix A, where the identification of Agent 1 is detailed as an example.

Remark 7. *Following this approach, the constraints imposed by the node equations are assigned to a given agent and are always satisfied. This fact represents an advantage with respect to the partitioning performed in [13], where the agents do not satisfy in general the node equations and the resulting components need to communicate partially following a hierarchical structure and generating virtual demands among them, i.e., they cannot work in a truly decentralized fashion.*

From an overall centralized viewpoint, the following equations are satisfied:

$$\mathbf{x}_{\mathcal{N}}(k+1) = \mathbf{A}_{\mathcal{N}}\mathbf{x}_{\mathcal{N}}(k) + \mathbf{B}_{\mathcal{N}}\mathbf{u}_{\mathcal{N}}(k) + \mathbf{B}_{\mathbf{p}_{\mathcal{N}}}\mathbf{d}_{\mathcal{N}}(k), \tag{43a}$$

$$0 = \mathbf{E}_{\mathcal{N}}\mathbf{u}_{\mathcal{N}}(k) + \mathbf{E}_{\mathbf{d}\mathcal{N}}\mathbf{d}_{\mathcal{N}}(k), \tag{43b}$$

with $\mathbf{x}_{\mathcal{N}}(k) \in \mathbb{R}^{63}$, $\mathbf{u}_{\mathcal{N}}(k) \in \mathbb{R}^{114}$ and $\mathbf{d}_{\mathcal{N}}(k) \in \mathbb{R}^{88}$. This way, (43a) corresponds with the dynamics of the storage tanks, and (43b) describes the network static relations due to the mass balance at each of the 17 junction nodes (see Fig. 3), with $\mathbf{E}_{\mathcal{N}} \in \mathbb{R}^{17\times 114}$ and $\mathbf{E}_{\mathbf{d}\mathcal{N}} \in \mathbb{R}^{17\times 88}$ being weighting matrices of proper dimensions.

Finally, consider the main physical constraints of the DWN given by the variables related to the tank volumes and manipulated flows, i.e., $\forall k$

$$\begin{aligned} \mathbf{x}_{\mathcal{N}}^{\min} \leq \mathbf{x}_{\mathcal{N}}(k) \leq \mathbf{x}_{\mathcal{N}}^{\max}, \\ \mathbf{u}_{\mathcal{N}}^{\min} \leq \mathbf{u}_{\mathcal{N}}(k) \leq \mathbf{u}_{\mathcal{N}}^{\max}. \end{aligned} \tag{44}$$

Remark 8. *Soft constraints have been introduced to implement the state constraints in (44). This fact, combined with the pre-partitioning based on the node equations, avoid infeasibility issues when solving the optimization problems.*

Remark 9. *In order to assess the impact of the links from a decision-making viewpoint, it is assumed that each agent has access to overall state $\mathbf{x}_{\mathcal{N}}(k)$, and knows how its decisions affect the overall system. However, decisions can only be coordinated inside coalitions, i.e., even when the different coalitions try to optimize the overall system performance, they cannot agree upon the value of the system variables.*

B. CONTROL OBJECTIVE: SYSTEM MANAGEMENT CRITERIA

The following management policies for the Barcelona DWN are considered given the knowledge of the system and the performance objectives to be reached (see [13], [14] for details):

- Minimizing drinking water production and transport costs due to chemicals, legal canons and electricity costs, which are expressed as

$$f_{1,i}(k) = (\mathbf{W}_{\mathbf{e_1}}\boldsymbol{\alpha_1} + \mathbf{W}_{\mathbf{e_2}}\boldsymbol{\alpha_2}(k))^{\mathrm{T}}\mathbf{u}_i(k), \tag{45}$$

  where vector $\boldsymbol{\alpha_1} \in \mathbb{R}^{n_{\mathrm{u}_i}}$ corresponds to water costs, vector $\boldsymbol{\alpha_2}(k) \in \mathbb{R}^{n_{\mathrm{u}_i}}$ considers time-dependent electricity costs, and matrices $\mathbf{W}_{\mathbf{e_1}}, \mathbf{W}_{\mathbf{e_2}} \in \mathbb{R}^{n_{\mathrm{u}_i}\times n_{\mathrm{u}_i}}$ add the corresponding prioritization to the aforementioned costs within the related multi-objective optimization problem.

- Maintaining the stored volume around a given safety value in case of emergency, which is achieved by minimizing

$$f_{2,i}(k) = \boldsymbol{\varsigma}_i^{\mathrm{T}}(k)\mathbf{W}_{\mathbf{x}}\boldsymbol{\varsigma}_i(k), \tag{46}$$

  with

$$\mathbf{x}_i(k) \geq \mathbf{x}_i^{\mathrm{saf}} - \boldsymbol{\varsigma}_i(k) \geq 0, \tag{47}$$

  where $\mathbf{x}_i^{\mathrm{saf}} \in \mathbb{R}^{n_{\mathrm{x}_i}}$ is a vector of safety volume thresholds in m^3 (conveniently determined according to the management company policies related to the DWN), with $\boldsymbol{\varsigma}_i(k) \in \mathbb{R}^{n_{\mathrm{x}_i}}$ representing the amount of volume going down from the desired thresholds, and where $\mathbf{W}_{\mathbf{x}} \in \mathbb{R}^{n_{\mathrm{x}_i}\times n_{\mathrm{x}_i}}$ is a weighting matrix.

- Penalizing sudden variations of the control inputs by minimizing

$$f_{3,i}(k) = \Delta\mathbf{u}_i^{\mathrm{T}}(k)\mathbf{R}_{\boldsymbol{\Delta}\mathbf{u}}\Delta\mathbf{u}_i(k), \tag{48}$$

  where $\Delta\mathbf{u}_i(k) = \mathbf{u}_i(k) - \mathbf{u}_i(k-1)$, and with $\mathbf{R}_{\boldsymbol{\Delta}\mathbf{u}} \in \mathbb{R}^{n_{\mathrm{u}_i}\times n_{\mathrm{u}_i}}$ also being a weighting matrix.

Hence, the individual cost related to agent $i \in \mathcal{N}$ that is considered in this paper is given by

$$\ell_i(\mathbf{x}_i(k), \mathbf{u}_i(k)) = f_{1,i}(k) + f_{2,i}(k) + f_{3,i}(k). \tag{49}$$

Finally, the aggregate cost of a certain communication component $\mathcal{C}$ is defined by

$$\ell_{\mathcal{C}}(\mathbf{x}_{\mathcal{C}}(k), \mathbf{u}_{\mathcal{C}}(k)) = \sum_{i\in\mathcal{C}} \ell_i(\mathbf{x}_i(k), \mathbf{u}_i(k)), \tag{50}$$

given that no couplings on the cost are considered in this work.

Remark 10. *Considering how (50) is built, (8) results in a quadratic programming (QP) problem. Therefore, convexity is guaranteed in the proposed approach, which allows a fast calculation of the solution for each optimization problem and a computation of (9) in polynomial time, as required in [28].*

VI. SIMULATION RESULTS

The Shapley-value-based partitioning algorithm presented in this paper has been tested for the Barcelona DWN by using the Matlab® solver `quadprog` in a 3.4 GHz Intel Octa-Core® i7-6400, 16 GB RAM computer. This way, a coalitional MPC scheme has been implemented in closed loop by considering $T_{\mathrm{sim}} = 24$ simulation instants (one day), and with $N_{\mathrm{p}} = 12$. The numerical values of the performance parameters are determined by a trial-and-error procedure, resulting in $\mathbf{W}_{\mathbf{e_1}} = 0.9\mathbf{I}$, $\mathbf{W}_{\mathbf{e_2}} = 0.5\mathbf{I}$, $\mathbf{W}_{\mathbf{x}} = 10\mathbf{I}$, $\mathbf{R}_{\boldsymbol{\Delta}\mathbf{u}} = 0.1\mathbf{I}$, with $\mathbf{I}$ being the identity matrix of suitable dimensions. Note also that there is no reference when considering (46), i.e., the controller chooses the most appropriate water volumes that satisfy the soft constraints imposed by $\mathbf{x}_i^{\mathrm{saf}}$. Likewise, the initial state is constant for all possible coalitions and is slightly above the minimum safety level.

The Shapley values for the 43 agents cannot be directly computed due to computational issues, which are solved here via the randomized method [28] introduced in Section III-A, considering $\varepsilon = 0.2\sigma_{\phi_i}$, $\forall i \in \mathcal{N}$, $\lambda = 0.1$, and $Z_{\lambda/2} = 1.6449$, which, in order to verify (20), requires a sample $\mathcal{Q}$ with $q = 68$ permutation vectors. According to (3), the

TABLE 2. Indices $\widetilde{\xi}_l(\mathcal{N}, \boldsymbol{v})$ for the 20 best/worst performance links

Best performance links $(\times 10^7)$		Worst performance links $(\times 10^7)$	
$\widetilde{\xi}_{1,2} = -5.9125$	$\widetilde{\xi}_{2,29} = -4.5847$	$\widetilde{\xi}_{6,40} = 1.7037$	$\widetilde{\xi}_{23,28} = 2.1592$
$\widetilde{\xi}_{1,3} = -4.3096$	$\widetilde{\xi}_{2,32} = -4.9091$	$\widetilde{\xi}_{8,23} = 1.9146$	$\widetilde{\xi}_{23,33} = 2.0847$
$\widetilde{\xi}_{1,38} = -5.0882$	$\widetilde{\xi}_{2,34} = -4.5568$	$\widetilde{\xi}_{8,40} = 1.9212$	$\widetilde{\xi}_{23,39} = 2.0647$
$\widetilde{\xi}_{2,3} = -7.5647$	$\widetilde{\xi}_{2,36} = -4.2433$	$\widetilde{\xi}_{9,40} = 1.7042$	$\widetilde{\xi}_{23,40} = 2.4991$
$\widetilde{\xi}_{2,4} = -4.5844$	$\widetilde{\xi}_{2,37} = -6.8329$	$\widetilde{\xi}_{12,23} = 2.0487$	$\widetilde{\xi}_{28,33} = 1.7513$
$\widetilde{\xi}_{2,5} = -4.8372$	$\widetilde{\xi}_{2,38} = -8,3433$	$\widetilde{\xi}_{12,28} = 1.7153$	$\widetilde{\xi}_{28,39} = 1.7313$
$\widetilde{\xi}_{2,15} = -4.5837$	$\widetilde{\xi}_{2,42} = -4.7561$	$\widetilde{\xi}_{12,40} = 2.0553$	$\widetilde{\xi}_{28,40} = 2.1658$
$\widetilde{\xi}_{2,16} = -4.5845$	$\widetilde{\xi}_{3,37} = -5.2300$	$\widetilde{\xi}_{20,23} = 2.0760$	$\widetilde{\xi}_{31,40} = 1.7007$
$\widetilde{\xi}_{2,24} = -4.5836$	$\widetilde{\xi}_{3,38} = -6.7404$	$\widetilde{\xi}_{20,28} = 1.7426$	$\widetilde{\xi}_{33,40} = 2.0913$
$\widetilde{\xi}_{2,27} = -4.5896$	$\widetilde{\xi}_{37,38} = -6.0086$	$\widetilde{\xi}_{20,40} = 2.0826$	$\widetilde{\xi}_{39,40} = 2.0713$

43 agents are related to 903 possible communication links. Likewise, the redistribution of the estimated Shapley value of the agent-based game to the links is obtained by using (11). Notice that for the estimation of the Shapley value, $43{\cdot}q$ coalitions have been evaluated by means of (9). The performance of these coalitions in terms of their cardinality is represented in Fig. 5, where it can be seen a low correlation between cardinality and coalition performance. In other words, the key of the partitioning performance is not related to group as many agents as possible but to select the clusters of cooperating agents properly.

The corresponding index $\widetilde{\xi}_l(\mathcal{N}, \boldsymbol{v})$ for every link $l = \{i, j\}$ is represented in a color scale in Fig. 6. Likewise, the values for the 20 best/worst performance links are explicitly indicated in Table 2.

A. SHAPLEY-VALUE-BASED PARTITIONING APPROACH (SVBA)

The proposed partitioning approach has been tested with the following thresholds, which have been determined by a trial-and-error tuning procedure:

$$\mathscr{L}_\mathrm{c} = 1.0 \times 10^7, \tag{51a}$$

$$\gamma = 2.5, \quad \nu = 2.3, \quad \rho = 100, \tag{51b}$$

$$\mathscr{L}_\mathrm{e} = 1.5 \times 10^7, \tag{52}$$

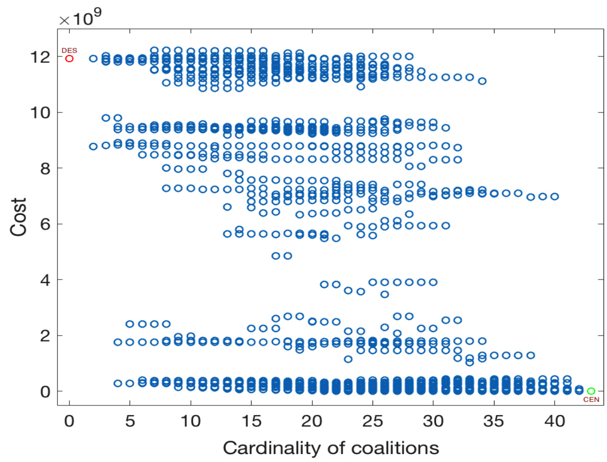

FIGURE 5. Cost of the coalitions needed by the randomized method in terms of their cardinality. Note that there are coalitions with few agents and proper performance – closer to the grand coalition – and vice versa, i.e., coalitions with many agents and performance similar to the empty coalition.

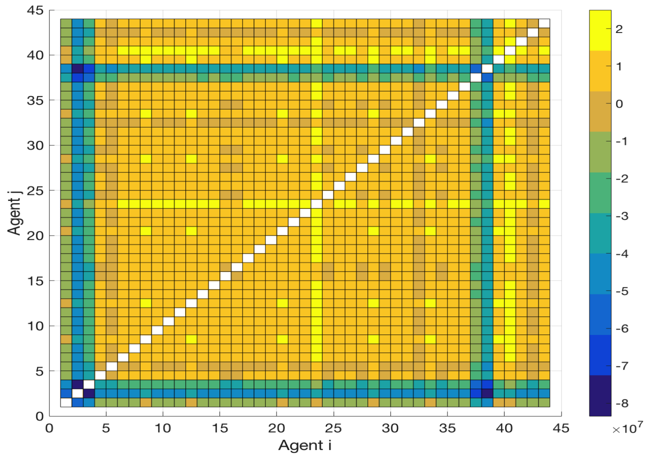

FIGURE 6. Estimated redistributions of the Shapley value. Yellowest and bluest colors represent most expensive and cheapest links, respectively. It can be seen that agents 2, 3 and 38 are endpoints of the links with best performance, whereas agents 23 and 40 are related to the links with the worst ones.

where several requirements have been considered in the adjustment of the aforementioned thresholds. In the first place, a cardinality constraint for any component of $0.2|\mathcal{N}|$ has been imposed, for our primary goal is to avoid components larger than one fifth of the system agents. Secondary goals were also considered, specifically to increase the cardinality of the resulting components so as to reduce the number of singletons.

Notice that $\mathcal{E}_\mathrm{e}$ is completely delimited by $\mathscr{L}_\mathrm{e}$. Likewise, with parameters in (51b) set to zero, set $\mathcal{E}_\mathrm{c}$ would also be completely determined by $\mathscr{L}_\mathrm{c}$. Under this premise, it is possible to represent the cardinality of both sets $\mathcal{E}_\mathrm{c}$ and $\mathcal{E}_\mathrm{e}$ as a function of any threshold corresponding to either $\mathscr{L}_\mathrm{c}$ or $\mathscr{L}_\mathrm{e}$, which is depicted in Fig. 7. Notice that both functions are symmetric given that in the limit case, i.e., $\mathscr{L}_\mathrm{c} = \mathscr{L}_\mathrm{e} = \mathscr{L}$, it is trivially verified

$$|\mathcal{E}_\mathrm{c}(\mathscr{L})| + |\mathcal{E}_\mathrm{e}(\mathscr{L})| = |\mathcal{E}|. \tag{53}$$

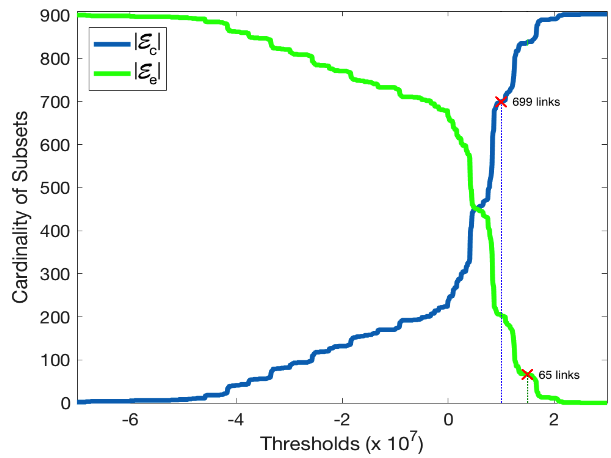

FIGURE 7. Cardinality of sets $\mathcal{E}_\mathrm{c}$ and $\mathcal{E}_\mathrm{e}$ as a function of a given threshold $\mathscr{L}$, for the particular case of not considering parameters to balance the size of the resulting components, i.e., $\gamma = \rho = 0$. The red crosses refer to the chosen thresholds, i.e., $|\mathcal{E}_\mathrm{c}(\mathscr{L}_\mathrm{c})| = 699$ and $|\mathcal{E}_\mathrm{e}(\mathscr{L}_\mathrm{e})| = 65$.

As seen in Fig. 7, set $\mathcal{E}_{\mathrm{c}}$ for $\gamma = \rho = 0$ would be composed of 699 links. Note that these links connect all agents in $\mathcal{N}$, achieving the grand coalition, i.e., $\mathcal{N}_{\mathcal{C}} = \{\mathcal{N}\}$. For this reason, (51b) is considered to penalize not only incoming links in $\mathcal{E}_{\mathrm{c}}(s)$, but also their remaining neighbors in $\mathcal{E}\backslash\mathcal{E}_{\mathrm{c}}(s)$. As a result, the following 52 links have been obtained and drawn in Fig. 8 in a color scale between green and yellow, with darkest links representing the useful ones:

$$\begin{aligned}\mathcal{E}_{\mathrm{c}} \;=\; \{&\{2,38\},\{2,3\},\{2,37\},\{3,38\},\{37,38\},\{1,2\},\{3,37\},\\ &\{1,38\},\{2,32\},\{2,5\},\{2,42\},\{2,27\},\{1,3\},\{32,38\},\\ &\{5,38\},\{38,42\},\{27,38\},\{1,37\},\{3,32\},\{3,5\},\{3,42\},\\ &\{3,27\},\{32,37\},\{5,37\},\{37,42\},\{27,37\},\{1,32\},\{1,5\},\\ &\{1,42\},\{1,27\},\{5,32\},\{32,42\},\{5,42\},\{16,29\},\{4,15\},\\ &\{24,34\},\{27,32\},\{4,29\},\{15,16\},\{4,16\},\{15,29\},\\ &\{5,27\},\{27,42\},\{24,36\},\{34,36\},\{21,43\},\{13,14\},\\ &\{18,26\},\{17,30\},\{10,19\},\{7,22\},\{11,25\}\},\end{aligned} \tag{54}$$

obtaining therefore the corresponding partitioning approach described below:

$$\begin{aligned}\mathcal{N}_{\mathcal{C}} = \{\, &\overbrace{\{1,2,3,5,27,32,37,38,42\}}^{\mathcal{C}_1}, \overbrace{\{4,15,16,29\}}^{\mathcal{C}_2},\\ &\overbrace{\{24,34,36\}}^{\mathcal{C}_3}, \overbrace{\{7,22\}}^{\mathcal{C}_4}, \overbrace{\{10,19\}}^{\mathcal{C}_5}, \overbrace{\{11,25\}}^{\mathcal{C}_6}, \overbrace{\{13,14\}}^{\mathcal{C}_7},\\ &\overbrace{\{17,30\}}^{\mathcal{C}_8}, \overbrace{\{18,26\}}^{\mathcal{C}_9}, \overbrace{\{21,43\}}^{\mathcal{C}_{10}}, \overbrace{\{6\}}^{\mathcal{C}_{11}}, \overbrace{\{8\}}^{\mathcal{C}_{12}}, \overbrace{\{9\}}^{\mathcal{C}_{13}}, \overbrace{\{12\}}^{\mathcal{C}_{14}}, \overbrace{\{20\}}^{\mathcal{C}_{15}},\\ &\overbrace{\{23\}}^{\mathcal{C}_{16}}, \overbrace{\{28\}}^{\mathcal{C}_{17}}, \overbrace{\{31\}}^{\mathcal{C}_{18}}, \overbrace{\{33\}}^{\mathcal{C}_{19}}, \overbrace{\{35\}}^{\mathcal{C}_{20}}, \overbrace{\{39\}}^{\mathcal{C}_{21}}, \overbrace{\{40\}}^{\mathcal{C}_{22}}, \overbrace{\{41\}}^{\mathcal{C}_{23}}\,\}.\end{aligned} \tag{55}$$

Likewise, $\mathcal{E}_{\mathrm{e}}$ is composed by the following 65 links:

$$\begin{aligned}\mathcal{E}_{\mathrm{e}} \;=\; \{&\{23,40\},\{28,40\},\{23,28\},\{33,40\},\{23,33\},\{20,40\},\\ &\{20,23\},\{39,40\},\{23,39\},\{12,40\},\{12,23\},\{8,40\},\\ &\{8,23\},\{28,33\},\{20,28\},\{28,39\},\{12,28\},\{9,40\},\\ &\{6,40\},\{31,40\},\{9,23\},\{6,23\},\{40,41\},\{23,31\},\\ &\{23,41\},\{35,40\},\{25,40\},\{11,40\},\{22,40\},\{23,35\},\\ &\{7,40\},\{10,40\},\{19,40\},\{17,40\},\{30,40\},\{18,40\},\\ &\{26,40\},\{20,33\},\{23,25\},\{14,40\},\{11,23\},\{22,23\},\\ &\{7,23\},\{10,23\},\{19,23\},\{17,23\},\{23,30\},\{18,23\},\\ &\{23,26\},\{13,40\},\{14,23\},\{33,39\},\{13,23\},\{20,39\},\\ &\{12,33\},\{12,20\},\{21,40\},\{12,39\},\{21,23\},\{40,43\},\\ &\{36,40\},\{23,43\},\{23,36\},\{8,28\},\{8,33\}\},\end{aligned} \tag{56}$$

and the corresponding set $\mathcal{E}_{\mathrm{e}_{\mathcal{C}}}$ is given by

$$\begin{aligned}\mathcal{E}_{\mathrm{e}_{\mathcal{C}}} \;=\; \{&\{\mathcal{C}_{14},\mathcal{C}_{15}\},\{\mathcal{C}_{14},\mathcal{C}_{21}\},\{\mathcal{C}_{15},\mathcal{C}_{21}\},\{\mathcal{C}_{16},\mathcal{C}_h\}_{h\geq 4\backslash\{16\}},\\ &\{\mathcal{C}_{17},\mathcal{C}_h\}_{h=\{12,14,15,19,21\}},\{\mathcal{C}_{19},\mathcal{C}_h\}_{h=\{12,14,15,21\}},\\ &\{\mathcal{C}_{22},\mathcal{C}_h\}_{h\geq 4\backslash\{22\}}\}.\end{aligned} \tag{57}$$

Finally, the overall network after performing the partitioning by the SVBA would be described by (55) and (57), achieving the network configuration specified in (33).

B. PARTITIONING APPROACH FINE TUNING (SVBA-FT)

The partitioning approach has been optimized for a simulation of a day with average demand and disturbances, starting by finding the partition with minimum cumulated cost $\mathcal{N}_{\mathcal{C}}^{(1)}$ from set $\Psi^{(0)}_{\mathcal{N},\mathcal{N}_{\mathcal{C}}} = \Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}}$, which in turns is obtained from solution $\mathcal{N}_{\mathcal{C}}$ described by (55). This optimization procedure has been recursively applied considering size constraints similar to those of the SVBA, i.e.,

TABLE 3. SVBA-FT Optimization Procedure

r	Switches with respect to (55)	$J_{\mathrm{cum}}(\mathcal{N}_{\mathcal{C}}^{(r)})$	$\gamma(r)$
1	Agent 43 from $\mathcal{C}_{10}$ to $\mathcal{C}_{17}$	9.7859×10^7	–
2	Agent 5 from $\mathcal{C}_1$ to $\mathcal{C}_2$	9.7096×10^7	–
3	Agent 43 from $\mathcal{C}_{17}$ to $\mathcal{C}_1$	2.5892×10^7	27.44 %
4	Agent 9 from $\mathcal{C}_{13}$ to $\mathcal{C}_2$	2.5098×10^7	25.73 %
5	Agent 11 from $\mathcal{C}_6$ to $\mathcal{C}_{14}$	2.4858×10^7	25.78 %
6	Agent 22 from $\mathcal{C}_4$ to $\mathcal{C}_{10}$	2.4619×10^7	1.66 %
7	Agent 17 from $\mathcal{C}_8$ to $\mathcal{C}_5$	2.4429×10^7	**0.90 %**

$$|\mathcal{C}_s| \leq \lceil 0.2|\mathcal{N}|\rceil = \lceil 8.6\rceil = 9, \quad \forall \mathcal{C}_s \in \mathcal{N}_{\mathcal{C}}^{(r)},\ s = 1,\ldots,n_{\mathrm{c}}, \tag{58}$$

and the following stopping criterion:

$$\gamma(r) < 1\%, \tag{59}$$

with

$$\gamma(r) = \frac{1}{3}\sum_{j=0}^{2}\left(\frac{J_{\mathrm{cum}}(\mathcal{N}_{\mathcal{C}}^{(r-3+j)}) - J_{\mathrm{cum}}(\mathcal{N}_{\mathcal{C}}^{(r-2+j)})}{J_{\mathrm{cum}}(\mathcal{N}_{\mathcal{C}}^{(r-3+j)})}\right), \tag{60}$$

where a 3-step average performance improvement has been considered, taking $J_{\mathrm{cum}}(\mathcal{N}_{\mathcal{C}}^{(0)}) = J_{\mathrm{cum}}(\mathcal{N}_{\mathcal{C}}) = 1.0662\times10^8$, $r \in \mathbb{N}^+, r > 3$.

The results related to every iteration are detailed in Table 3, where only $r_{\mathrm{stop}} = 7$ steps have been needed to fulfill the stopping criterion described by (59), which indicates that the solution in (55) is indeed a suitable starting point. Note that

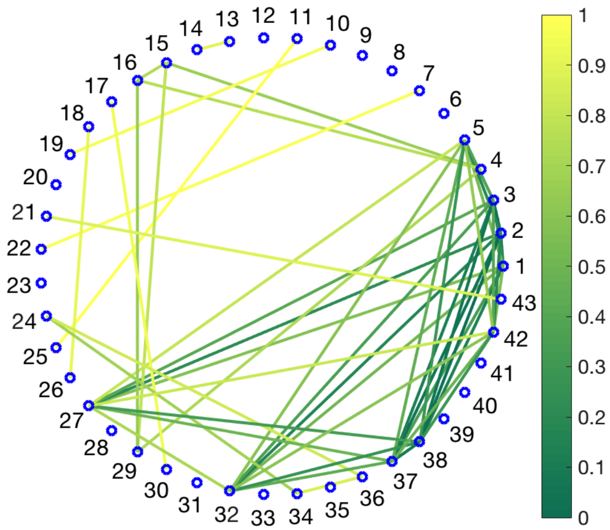

FIGURE 8. Links belonging to $\mathcal{E}_{\mathrm{c}}$ after applying the partitioning procedure, with their performance normalized. This way, value "0" refers to the link with best performance, i.e., $\{2,38\}$, and it is drawn in pure green, and value "1" is related to the link with worst performance out of the 52 links in $\mathcal{E}_{\mathrm{c}}$, i.e., $\{11,25\}$, and it is represented in pure yellow.

the number of partitions explored by the greedy procedure can be easily calculated by $\sum_{r=0}^{r_{\text{stop}}-1} |\Psi^{(r)}_{\mathcal{N},\mathcal{N}_{\mathcal{C}}}| = 6143$. The definitive solution after applying the optimization procedure is determined by

$$\begin{aligned}\mathcal{N}_{\mathcal{C}}^{\text{opt}} = \{ & \overbrace{\{1,2,3,27,32,37,38,42,43\}}^{\mathcal{C}_1^{\text{opt}}}, \overbrace{\{4,5,9,15,16,29\}}^{\mathcal{C}_2^{\text{opt}}}, \\ & \overbrace{\{10,17,19\}}^{\mathcal{C}_3^{\text{opt}}}, \overbrace{\{24,34,36\}}^{\mathcal{C}_4^{\text{opt}}}, \overbrace{\{11,12\}}^{\mathcal{C}_5^{\text{opt}}}, \overbrace{\{13,14\}}^{\mathcal{C}_6^{\text{opt}}}, \overbrace{\{18,26\}}^{\mathcal{C}_7^{\text{opt}}}, \\ & \overbrace{\{21,22\}}^{\mathcal{C}_8^{\text{opt}}}, \overbrace{\{6\}}^{\mathcal{C}_9^{\text{opt}}}, \overbrace{\{7\}}^{\mathcal{C}_{10}^{\text{opt}}}, \overbrace{\{8\}}^{\mathcal{C}_{11}^{\text{opt}}}, \overbrace{\{20\}}^{\mathcal{C}_{12}^{\text{opt}}}, \overbrace{\{23\}}^{\mathcal{C}_{13}^{\text{opt}}}, \overbrace{\{25\}}^{\mathcal{C}_{14}^{\text{opt}}}, \overbrace{\{28\}}^{\mathcal{C}_{15}^{\text{opt}}}, \overbrace{\{30\}}^{\mathcal{C}_{16}^{\text{opt}}}, \\ & \overbrace{\{31\}}^{\mathcal{C}_{17}^{\text{opt}}}, \overbrace{\{33\}}^{\mathcal{C}_{18}^{\text{opt}}}, \overbrace{\{35\}}^{\mathcal{C}_{19}^{\text{opt}}}, \overbrace{\{39\}}^{\mathcal{C}_{20}^{\text{opt}}}, \overbrace{\{40\}}^{\mathcal{C}_{21}^{\text{opt}}}, \overbrace{\{41\}}^{\mathcal{C}_{22}^{\text{opt}}} \}.\end{aligned} \tag{61}$$

Finally, note that the switches performed as a result of the optimization procedure do not imply to remove any communication link in set $\mathcal{E}_{\text{e}}$. Consequently, set $\mathcal{E}_{\text{e}\mathcal{C}}$ remains constant with respect to (57), being the final network configuration described by

$$(\mathcal{N}_{\mathcal{C}}^{\text{opt}}, \mathcal{E}_{\mathcal{C}}^{\text{opt}} \backslash \mathcal{E}_{\text{e}\mathcal{C}}), \tag{62}$$

with $\mathcal{E}_{\mathcal{C}}^{\text{opt}} = \mathcal{N}_{\mathcal{C}}^{\text{opt}} \times \mathcal{N}_{\mathcal{C}}^{\text{opt}}$. In any case, note that cheaper/more expensive agents illustrated in Fig. 6 are not affected by the changes introduced by the fine tuning.

Remark 11. *Given that the proposed procedure optimizes discrete variables, i.e., the components, $\gamma(r)$ might increase in further iterations with r. That is, there is no guarantee that the imposed limit becomes a bound for later iterations. Nevertheless, the average way in which $\gamma(r)$ is defined mitigates this possibility.*

C. COMPARISON OF THE PROPOSED PARTITIONING SOLUTIONS WITH OTHER SCHEMES IN THE LITERATURE

In this section, both the original and refined proposed partitioning solutions will be compared with the solutions obtained by applying other different partitioning approaches [10], [13] to the full Barcelona DWN, and also with the centralized and decentralized configurations.

1) An alternative Barcelona DWN partitioning scheme

In [13], an alternative partitioning method of the same Barcelona DWN model analyzed in this paper, which basically consists in a graph-theory-based approach (GTBA), is proposed. Nevertheless, that work follows a different criterion to define the agents, and for this reason their results are not directly comparable with the approach proposed here. With the aim of providing a way to compare all approaches, it has been considered that each of the 43 agents defined here belong to a component described in [13] if all its related variables, i.e., states, inputs, disturbances, are contained into this component. Under this assumption, which represents an approximation of the partitioning in [13], most of the 43 agents have been distributed into the six components in [13], with the exception of agents 1 and 2, which have been assumed to belong to new independent components. Taking this fact into account, the partitioning provided by the GTBA can be modeled by

$$\begin{aligned}\mathcal{N}_{\mathcal{C}}^{\text{GTBA}} = \{ & \overbrace{\{24,25,26,27,28,37,38,42,43\}}^{\mathcal{C}_1^{\text{GTBA}}}, \\ & \overbrace{\{10,11,12,13,14,15,16,17,18,19,20\}}^{\mathcal{C}_2^{\text{GTBA}}}, \\ & \overbrace{\{29,30,31,32\}}^{\mathcal{C}_3^{\text{GTBA}}}, \overbrace{\{33,34,35,36\}}^{\mathcal{C}_4^{\text{GTBA}}}, \overbrace{\{21,22,23\}}^{\mathcal{C}_5^{\text{GTBA}}}, \\ & \overbrace{\{3,4,5,6,7,8,9,39,40,41\}}^{\mathcal{C}_6^{\text{GTBA}}}, \overbrace{\{1\}}^{\mathcal{C}_7^{\text{GTBA}}}, \overbrace{\{2\}}^{\mathcal{C}_8^{\text{GTBA}}} \},\end{aligned} \tag{63}$$

where it can be seen that this solution does not respect the size constraints imposed to the SVBA/SVBA-FT, which represents an advantage for the GTBA.

2) A partitioning approach applied to other DWN

In [10], a partitioning approach based on social network algorithms (SNBA) is introduced and applied to the Parete DWN, located in the South of Caserta, Italy. Basically, this approach uses the centrality index called *edge betweenness* as a metric to identify the boundaries of communities [39]. Let $(\mathcal{V}, \mathcal{L})$ be a directed graph that describes the direction of the flows, symbolized by links $l_{\text{f}} \in \mathcal{L}$, among any pair of vertices $\{s,t\} \in \mathcal{V}$, which are related to water entities, e.g., tanks. The edge betweenness $c_{\text{B}}(l_{\text{f}})$ of a link l_{f} is defined as the number of optimal paths between vertex pairs that run along link l_{f}, summed over all vertex pairs, as follows [40]:

$$c_{\text{B}}(l_{\text{f}}) = \sum_{\{s,t\}\in\mathcal{V}} \frac{\sigma(s,t|l_{\text{f}})}{\sigma(s,t)}, \tag{64}$$

where $\sigma(s,t)$ is the number of shortest (s,t)-paths, and $\sigma(s,t|l_{\text{f}})$ is the number of those paths passing through link l_{f}. This way, an optimal community cluster can be defined by progressively removing edges with high value of edge betweenness from the original graph [39]. In other words, index (64) identifies edges in a network that lie between communities, which can be progressively removed leaving behind just the communities themselves.

Note that for the Barcelona DWN case study, the direction of the flows among the *water tanks and junction nodes* is described by graph in Fig. 4. Given that in this work a pre-partitioning into agents has been made, to properly apply the scheme proposed in [10] it is needed to map the previous graph into a new one that depicts the direction of the flows among the *agents*. As a result, graph $(\mathcal{V}, \mathcal{L})$ represented in Fig. 9, with $\mathcal{V} = \mathcal{N} = 43$ agents and $\mathcal{L} = 49$ directed links has been obtained, where each agent comprises information regarding several tanks and nodes following the criterion established in Section V-A. Once graph $(\mathcal{V}, \mathcal{L})$ is established,

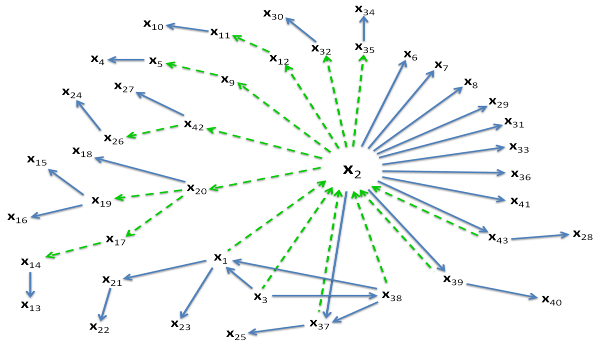

FIGURE 9. Graph $(\mathcal{V}, \mathcal{L})$ representing the direction of the flows among the 43 agents for the full Barcelona DWN. As it can be seen, agent 2 has a strong centrality regarding the flows.

it is possible to compute index (64) for all these 49 directed links. Finally, in order to obtain the partitioning solution, the links with higher edge betweenness have been progressively removed until achieve the stopping criterion of a maximum cardinality for any component of $0.2|\mathcal{N}|$, as done in the partitioning approach proposed in this paper. As a result, 18 links have been removed, which are depicted by green dashed arrows in Fig. 9, and explicitly represented joint to their $c_{\mathrm{B}}(l_{\mathrm{f}})$ in Table 4. The resulting partitioning scheme is given below:

$$\begin{aligned}\mathcal{N}_{\mathcal{C}}^{\mathrm{SNBA}} = \{ & \overbrace{\{2,6,7,8,29,31,33,36,41\}}^{c_1^{\mathrm{SNBA}}}, \\ & \overbrace{\{1,3,21,22,23,25,37,38\}}^{c_2^{\mathrm{SNBA}}}, \overbrace{\{15,16,19\}}^{c_3^{\mathrm{SNBA}}}, \overbrace{\{4,5\}}^{c_4^{\mathrm{SNBA}}}, \\ & \overbrace{\{10,11\}}^{c_5^{\mathrm{SNBA}}}, \overbrace{\{13,14\}}^{c_6^{\mathrm{SNBA}}}, \overbrace{\{18,20\}}^{c_7^{\mathrm{SNBA}}}, \overbrace{\{24,26\}}^{c_8^{\mathrm{SNBA}}}, \overbrace{\{27,42\}}^{c_9^{\mathrm{SNBA}}}, \\ & \overbrace{\{28,43\}}^{c_{10}^{\mathrm{SNBA}}}, \overbrace{\{30,32\}}^{c_{11}^{\mathrm{SNBA}}}, \overbrace{\{34,35\}}^{c_{12}^{\mathrm{SNBA}}}, \overbrace{\{39,40\}}^{c_{13}^{\mathrm{SNBA}}}, \\ & \overbrace{\{9\}}^{c_{14}^{\mathrm{SNBA}}}, \overbrace{\{12\}}^{c_{15}^{\mathrm{SNBA}}}, \overbrace{\{17\}}^{c_{16}^{\mathrm{SNBA}}} \}. \end{aligned} \tag{65}$$

Remark 12. *Directed graph $(\mathcal{V}, \mathcal{L}) = (\mathcal{N}, \mathcal{L})$ has nothing to do with complete undirected graph $(\mathcal{N}, \mathcal{E})$, which was used in the partitioning algorithm proposed in Section IV. The former represents the direction of the physical flows among the agents, i.e., $|\mathcal{L}| = 49$, whereas the latter assumes that all agents are initially interconnected, i.e., $|\mathcal{E}| = 903$, to later classify these undirected communication links into sets $\mathcal{E}_{\mathrm{c}}$ and $\mathcal{E}_{\mathrm{e}}$ regarding their control performance.*

TABLE 4. Edge betweenness $c_{\mathrm{B}}(l_{\mathrm{f}})$ of directed links l_{f} removed from graph in Fig. 9 to find communities

$c_{\mathrm{B}}(l_{\mathrm{f}})$	l_{f}	$c_{\mathrm{B}}(l_{\mathrm{f}})$	l_{f}	$c_{\mathrm{B}}(l_{\mathrm{f}})$	l_{f}
56	$\{2, 20\}$	35	$\{43, 2\}$	18	$\{17, 14\}$
37	$\{1, 2\}$	28	$\{2, 42\}$	16	$\{9, 5\}$
36	$\{3, 2\}$	24	$\{20, 17\}$	16	$\{12, 11\}$
35	$\{37, 2\}$	24	$\{20, 19\}$	16	$\{42, 26\}$
35	$\{38, 2\}$	21	$\{2, 9\}$	14	$\{2, 32\}$
35	$\{39, 2\}$	21	$\{2, 12\}$	14	$\{2, 35\}$

TABLE 5. Sensitivity Analysis of the Different Schemes

Indices	CEN	SVBA-FT	SVBA
$J_{\mathrm{cum}}(\mathcal{N}_{\mathcal{C}})$	2.1405×10^6	2.4429×10^7	1.0662×10^8
$\lvert\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}}\rvert$	43	841	898
$J_{\mathrm{cum}}^{\max}(\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}})$	2.8109×10^8	2.6106×10^8	2.7493×10^8
$J_{\mathrm{cum}}^{\mu}(\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}})$	1.3750×10^7	3.6118×10^7	1.1541×10^8
$J_{\mathrm{cum}}^{\min}(\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}})$	2.1405×10^6	2.4355×10^7	9.7859×10^7
$\eta_{0.1\%}$	100.00 %	98.22 %	97.10 %
Indices	**SNBA [10]**	**GTBA [13]**	**DEC**
$J_{\mathrm{cum}}(\mathcal{N}_{\mathcal{C}})$	2.9002×10^8	1.1120×10^{10}	1.1925×10^{10}
$\lvert\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}}\rvert$	682	341	903
$J_{\mathrm{cum}}^{\max}(\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}})$	8.7937×10^9	1.1263×10^{10}	1.2231×10^{10}
$J_{\mathrm{cum}}^{\mu}(\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}})$	6.3178×10^8	1.0855×10^{10}	1.1884×10^{10}
$J_{\mathrm{cum}}^{\min}(\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}})$	1.0748×10^8	2.8767×10^8	2.9576×10^8
$\eta_{0.1\%}$	89.88 %	86.22 %	97.79 %

Sensitivity analysis comparison

All approaches have been tested by using the sensitivity analysis introduced in Section IV-A, where the solutions, defined respectively by (55), (61), (63), (65), and also the centralized (CEN) and decentralized (DEC) configurations, have been compared with those in their corresponding sets $\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}}$. The cardinality of these sets (obtained by (38)) and the related indices are illustrated in Table 5 for the same average day used in the previous section, where an uncertainty limit of $p = 0.1\%$ has been assumed in the computation of η_p. As expected, the value $J_{\mathrm{cum}}^{\min}(\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}})$ of the SVBA corresponds with the first step of the optimization procedure in Table 3.

Note that $|\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}}|$ provides the number of partitioning solutions explored in the sensitivity analysis. Nevertheless, only the solutions of that set that satisfy the size constraints, i.e., maximum cardinality for any component of $0.2|\mathcal{N}|$, should be considered. This way, note that the CEN has been included in this comparison even without any element in the corresponding set $\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}}$ trivially satisfying this size constraint, which explains its best performance. Likewise, as commented before, there are two components of the GTBA that also do not respect the size constraints, which represent an advantage to this scheme in the comparisons.

As can be seen, the values of $J_{\mathrm{cum}}(\mathcal{N}_{\mathcal{C}})$, $J_{\mathrm{cum}}^{\min}(\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}})$ and $J_{\mathrm{cum}}^{\max}(\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}})$ for the SVBA/SVBA-FT improve those of the GTBA, SNBA and DEC. That is, the proposed solutions improve the performance of the rest. According to η_p, it can be checked that the SVBA outperforms a higher portion of partitioning solutions in corresponding sets $\Psi_{\mathcal{N},\mathcal{N}_{\mathcal{C}}}$ than the GTBA and SNBA, and is in the order of the DEC, which in any case is improved by the SVBA-FT. This finding is consistent with Fig. 5, which already illustrated that only a few topological changes in the network can increase the performance substantially. Note as well that the SVBA/SVBA-FT also outperform the other methods when the difference between the cost of the different approaches and the cor-

TABLE 6. Overview of the different methods considered

Scheme	$\lvert\mathcal{N}_\mathcal{C}\rvert$	$J_{\text{cum}}(\mathcal{N}_\mathcal{C})$-30 days
CEN	1	3.7569×10^6
SVBA-FT	22	1.7422×10^8
SVBA	23	7.7357×10^8
SNBA [10]	16	1.3220×10^{11}
GTBA [13]	8	1.1357×10^{12}
DEC	43	2.3669×10^{12}

responding minimum in their set of alternatives $\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}}$ is examined, i.e., the proposed approaches are closer to these minimum-cost solutions than the other methods. Likewise, it is interesting to check that both the SVBA and SVBA-FT have a better performance than the mean performance of that set, i.e., $J_{\text{cum}}(\mathcal{N}_\mathcal{C}) < J^{\mu}_{\text{cum}}(\Psi_{\mathcal{N},\mathcal{N}_\mathcal{C}})$, which is also the case of the SNBA but does not occur in the GTBA and DEC. Finally, notice that all parameters in the optimized scheme SVBA-FT improve those in the SVBA, as expected. All in all, these results indicate that both proposed approaches outperform the rest of schemes.

D. PARTITIONING LONG-SIMULATION OVERVIEW

A comparison between the different schemes considered in this paper is summarized in Table 6, where the cumulated cost of a 30-day simulation scenario with demand and disturbances taken from real data has been calculated for each approach. Note that it is reasonable to test the results in a longer scenario than the one used for the design, i.e., one day. It can be seen that both the SVBA and SVBA-FT improve the results of the GTBA, SNBA and DEC, showing the effectiveness of the partitioning algorithm proposed in this paper.

Notice that the SVBA provides us with a suboptimal solution within the set of B_{43} options (recall (37)), which represents a suitable starting point for the fine tuning. Then, the goal of the SVBA-FT is to increase the performance by carrying out a greedy search around SVBA. In particular, significant gains can be obtained by reducing any violation of constraints, which are severely penalized by soft constraints. Also, the fine tuning helps to mitigate possible deviations introduced by the randomized method used for the estimation of the Shapley value. In any case, note that the fact that of some links originally included in sets $\mathcal{E}_c/\mathcal{E}_e$ could be removed due to the SVBA-FT does not break any physical constraints of the original DWN given that these constraints are used to define the agents. This is not the case of other schemes such us [10], where a fine tuning procedure could break some physical/topological connections.

Finally, these partitioning results can be improved once the communication components start exchanging information, e.g., by using a coalitional control scheme [22], [24].

VII. CONCLUSIONS

In this paper, a heuristic partitioning algorithm for LSS has been introduced considering that agents are able to operate in a decentralized fashion. This method ranks the communication links inside a network from a control-performance perspective. A game over agents based on the utilization of coalitional MPC to control the plant has been considered, and a randomized method has been used to estimate their Shapley value to deal with combinatorial explosion issues. The redistribution of the estimated Shapley value of the agents among the links has been proposed as a measure of their relevance in the control system and to fix/remove them.

The proposed algorithm calculates sets $\mathcal{E}_c$ and $\mathcal{E}_e$. The links in the former set will determine the communication components, where several mechanisms to avoid the formation of large clusters of cooperating agents have been considered. Once the components are established, some connections among them are disconnected by using the information in the latter set, providing a proper configuration of the overall network. The resulting partitioning has also been refined by a method that optimizes it recursively, based on some cost indices and size constraints. Both the original and refined approaches have been tested with the Barcelona DWN as a case study, providing reasonable solutions that outperform other partitioning schemes available in the literature.

.

APPENDIX A STATES, INPUTS AND DISTURBANCES RELATED TO THE AGENTS

As mentioned in Section V-A, the flows belonging to the following node equations:

$$\begin{aligned} u(25) - \boldsymbol{u(27)} - \boldsymbol{u(28)} - \boldsymbol{u(29)} - \boldsymbol{u(105)} - d(15) &= 0, \\ u(27) - \boldsymbol{u(26)} - d(36) &= 0, \end{aligned} \tag{66}$$

are physically interconnected, in particular by means of flow $u(27)$. Then, all states that comprise *incoming* flows involved in (66), which are represented in blue color, should belong to the same agent, i.e.,

$$\begin{aligned} x^+(22) &= x(22) + u(22) + \boldsymbol{u(26)} - u(23) - d(23), \\ x^+(27) &= x(27) + \boldsymbol{u(28)} + \boldsymbol{u(29)} - u(30) - d(28), \\ x^+(57) &= x(57) + u(80) + u(89) + \boldsymbol{u(105)} - u(78), \end{aligned} \tag{67}$$

where superindex $^+$ refers to the successors state. Finally, note that outflows in the states – drawn in red color – are considered here as disturbances.

By considering this approach, the following 43 Agents have been obtained:

- **Agent 1:**
  - States: $x(22)$, $x(27)$, $x(57)$.
  - Inputs: $u(22)$, $u(25)$, $u(26)$, $u(27)$, $u(28)$, $u(29)$, $u(80)$, $u(89)$, $u(105)$.
  - Disturbances: $d(15)$, $d(23)$, $d(28)$, $d(36)$, $u(23)$, $u(30)$, $u(78)$.

- **Agent 2:**
  - States: $x(7)$, $x(8)$, $x(9)$, $x(10)$, $x(32)$, $x(35)$, $x(38)$, $x(42)$, $x(43)$, $x(44)$, $x(45)$, $x(46)$, $x(48)$, $x(49)$, $x(50)$, $x(53)$, $x(55)$, $x(56)$, $x(60)$.
  - Inputs: $u(7)$, $u(8)$, $u(9)$, $u(10)$, $u(33)$, $u(36)$, $u(40)$, $u(41)$, $u(44)$, $u(45)$, $u(46)$, $u(47)$, $u(48)$, $u(49)$, $u(50)$, $u(51)$, $u(52)$, $u(53)$, $u(54)$, $u(55)$, $u(56)$, $u(57)$, $u(58)$, $u(59)$, $u(61)$, $u(62)$, $u(63)$, $u(64)$, $u(66)$, $u(67)$, $u(70)$, $u(71)$, $u(72)$, $u(74)$, $u(75)$, $u(76)$, $u(77)$, $u(78)$, $u(79)$, $u(81)$, $u(82)$, $u(83)$, $u(84)$, $u(85)$, $u(86)$, $u(87)$, $u(90)$, $u(91)$, $u(92)$, $u(93)$, $u(95)$, $u(96)$, $u(97)$, $u(106)$, $u(107)$, $u(110)$, $u(111)$, $u(113)$, $u(114)$.
  - Disturbances: $d(7)$, $d(8)$, $d(9)$, $d(10)$, $d(32)$, $d(40)$, $d(42)$, $d(43)$, $d(45)$, $d(47)$, $d(50)$, $d(52)$, $d(53)$, $d(54)$, $d(55)$, $d(57)$, $d(58)$, $d(59)$, $d(60)$, $d(62)$, $d(63)$, $d(64)$, $d(65)$, $d(69)$, $d(70)$, $d(71)$, $d(72)$, $d(73)$, $d(74)$, $d(75)$, $d(76)$, $d(77)$, $d(79)$, $d(80)$, $d(81)$, $d(82)$, $d(83)$, $d(84)$, $u(3)$, $u(4)$, $u(5)$, $u(6)$, $u(13)$, $u(21)$, $u(38)$, $u(42)$, $u(43)$, $u(60)$, $u(67)$, $u(68)$, $u(69)$, $u(72)$, $u(74)$, $u(94)$, $u(95)$, $u(98)$, $u(101)$, $u(106)$, $u(107)$, $u(108)$, $u(109)$, $u(111)$, $u(112)$.
- **Agent 3:**
  - States: $x(59)$.
  - Inputs: $u(102)$, $u(103)$, $u(104)$.
  - Disturbances: $u(88)$, $u(89)$, $u(97)$.
- **Agent 4:**
  - States: $x(1)$.
  - Inputs: $u(1)$.
  - Disturbances: $d(1)$.
- **Agent 5:**
  - States: $x(2)$.
  - Inputs: $u(2)$.
  - Disturbances: $d(2)$, $u(1)$.
- **Agent 6:**
  - States: $x(3)$.
  - Inputs: $u(3)$.
  - Disturbances: $d(3)$.
- **Agent 7:**
  - States: $x(4)$.
  - Inputs: $u(4)$.
  - Disturbances: $d(4)$.
- **Agent 8:**
  - States: $x(5)$.
  - Inputs: $u(5)$.
  - Disturbances: $d(5)$.
- **Agent 9:**
  - States: $x(6)$.
  - Inputs: $u(6)$.
  - Disturbances: $d(6)$, $u(2)$.
- **Agent 10:**
  - States: $x(11)$.
  - Inputs: $u(11)$.
  - Disturbances: $d(11)$.
- **Agent 11:**
  - States: $x(12)$.
  - Inputs: $u(12)$.
  - Disturbances: $d(12)$, $u(11)$.
- **Agent 12:**
  - States: $x(13)$.
  - Inputs: $u(13)$.
  - Disturbances: $d(13)$, $u(12)$.
- **Agent 13:**
  - States: $x(14)$.
  - Inputs: $u(14)$.
  - Disturbances: $d(14)$.
- **Agent 14:**
  - States: $x(15)$.
  - Inputs: $u(15)$.
  - Disturbances: $d(16)$, $u(14)$.
- **Agent 15:**
  - States: $x(16)$.
  - Inputs: $u(16)$.
  - Disturbances: $d(17)$.
- **Agent 16:**
  - States: $x(17)$.
  - Inputs: $u(17)$.
  - Disturbances: $d(18)$.
- **Agent 17:**
  - States: $x(18)$.
  - Inputs: $u(18)$.
  - Disturbances: $d(19)$, $u(15)$.
- **Agent 18:**
  - States: $x(19)$.
  - Inputs: $u(19)$.
  - Disturbances: $d(20)$.
- **Agent 19:**
  - States: $x(20)$.
  - Inputs: $u(20)$.
  - Disturbances: $d(21)$, $u(16)$, $u(17)$.
- **Agent 20:**
  - States: $x(21)$.
  - Inputs: $u(21)$.
  - Disturbances: $d(22)$, $u(18)$, $u(19)$, $u(20)$.
- **Agent 21:**
  - States: $x(23)$.
  - Inputs: $u(23)$.
  - Disturbances: $d(24)$, $u(24)$.
- **Agent 22:**
  - States: $x(24)$.
  - Inputs: $u(24)$.
  - Disturbances: $d(25)$, $d(38)$.
- **Agent 23:**
  - States: $x(25)$.
  - Inputs: $u(30)$.
  - Disturbances: $d(26)$.
- **Agent 24:**
  - States: $x(26)$.
  - Inputs: $u(31)$.
  - Disturbances: $d(27)$, $d(35)$.
- **Agent 25:**
  - States: $x(30)$.
  - Inputs: $u(32)$.
  - Disturbances: $d(34)$.
- **Agent 26:**
  - States: $x(28)$.
  - Inputs: $u(34)$.
  - Disturbances: $d(29)$, $u(31)$.
- **Agent 27:**
  - States: $x(29)$.
  - Inputs: $u(35)$.
  - Disturbances: $d(30)$, $d(37)$.
- **Agent 28:**
  - States: $x(36)$.
  - Inputs: $u(37)$.
  - Disturbances: $d(41)$.

- **Agent 29:**
  - States: $x(34)$.
  - Inputs: $u(38)$.
  - Disturbances: $d(39)$.
- **Agent 30:**
  - States: $x(40)$.
  - Inputs: $u(39)$.
  - Disturbances: $d(51)$.
- **Agent 31:**
  - States: $x(39)$.
  - Inputs: $u(42)$.
  - Disturbances: $d(48)$, $d(49)$.
- **Agent 32:**
  - States: $x(41)$.
  - Inputs: $u(43)$.
  - Disturbances: $d(56)$, $u(39)$.
- **Agent 33:**
  - States: $x(47)$.
  - Inputs: $u(60)$.
  - Disturbances: $d(61)$.
- **Agent 34:**
  - States: $x(51)$.
  - Inputs: $u(65)$.
  - Disturbances: $d(66)$.
- **Agent 35:**
  - States: $x(52)$.
  - Inputs: $u(68)$.
  - Disturbances: $d(67)$, $u(65)$.
- **Agent 36:**
  - States: $x(54)$.
  - Inputs: $u(69)$.
  - Disturbances: $d(68)$.
- **Agent 37:**
  - States: $x(33)$.
  - Inputs: $u(73)$, $u(108)$.
  - Disturbances: $d(33)$, $u(32)$, $u(33)$.
- **Agent 38:**
  - States: $x(58)$.
  - Inputs: $u(88)$.
  - Disturbances: $d(78)$, $u(73)$, $u(80)$, $u(82)$, $u(87)$.
- **Agent 39:**
  - States: $x(61)$.
  - Inputs: $u(94)$, $u(98)$, $u(99)$.
  - Disturbances: $d(85)$, $u(93)$, $u(100)$.
- **Agent 40:**
  - States: $x(62)$.
  - Inputs: $u(100)$.
  - Disturbances: $d(87)$, $d(88)$.
- **Agent 41:**
  - States: $x(63)$.
  - Inputs: $u(101)$.
  - Disturbances: $d(86)$.
- **Agent 42:**
  - States: $x(31)$.
  - Inputs: $u(109)$.
  - Disturbances: $d(31)$, $u(34)$, $u(35)$.
- **Agent 43:**
  - States: $x(37)$.
  - Inputs: $u(112)$.
  - Disturbances: $d(44)$, $d(46)$, $u(37)$, $u(114)$.

ACKNOWLEDGEMENT

The authors would like to thank the Editor and the anonymous reviewers for their valuable suggestions and comments.

REFERENCES

[1] E. F. Camacho and C. Bordons, Model Predictive Control, 2^{nd} ed., ser. Advances Textbooks in Control and Signal Processing. London, England: Springer, 2003.

[2] J. M. Maestre and R. R. Negenborn, Eds., Distributed Model Predictive Control Made Easy, ser. Intelligent Systems, Control and Automation: Science and Engineering. Dordrecht, The Netherlands: Springer, 2014, vol. 69.

[3] R. R. Negenborn, P.-J. van Overloop, T. Keviczky, and B. De Schutter, "Distributed model predictive control of irrigation canals," Networks and Heterogeneous Media, vol. 4, no. 2, pp. 359–380, June 2009.

[4] D. D. Siljak, Decentralized Control of Complex Systems. Boston, Massachusetts, USA: Academic Press, 1991.

[5] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz, "Recent advances in graph partitioning," in Algorithm Engineering. Selected Results and Surveys, ser. Lecture Notes in Computer Science, L. Kliemann and P. Sanders, Eds. Springer, 2016, vol. 9220, ch. 25, pp. 117–158.

[6] H. Meyerhenke, P. Sanders, and C. Schulz, "Parallel graph partitioning for complex networks," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 9, pp. 2625–2638, September 2017.

[7] Y. Guo, S. Hong, H. Chafi, A. Iosup, and D. Epema, "Modeling, analysis, and experimental comparison of streaming graph-partitioning policies," Journal of Parallel and Distributed Computing, vol. 108, pp. 106–121, October 2017.

[8] F. Rahimian, A. H. Payberah, S. Girdzijauskas, M. Jelasity, and S. Haridi, "A distributed algorithm for large-scale graph partitioning," ACM Transactions on Autonomous and Adaptive Systems, vol. 10, no. 2, pp. 12:1–12:24, June 2015.

[9] R. Schneider, "A solution for the partitioning problem in partition-based-moving-horizon estimation," IEEE Transactions on Automatic Control, vol. 62, no. 6, pp. 3076–3082, June 2017.

[10] A. Di Nardo, M. Di Natale, C. Giudicianni, D. Musmarra, G. F. Santonastaso, and S. A., "Water distribution system clustering and partitioning based on social network algorithms," Procedia Engineering, vol. 119, pp. 196–205, September 2015.

[11] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," IEEE Transactions on Evolutionary Computation, vol. 16, no. 3, pp. 418–430, June 2012.

[12] K. Tsumura and H. Yamamoto, "Optimal multiple controlling nodes problem for multi-agent systems via Alt-PageRank," in Proceedings of the 4^{th} IFAC Workshop on Distributed Estimation and Control in Networked Systems, Koblenz, Germany, September 2013, pp. 433–438.

[13] C. Ocampo-Martinez, S. Bovo, and V. Puig, "Partitioning approach oriented to the decentralised predictive control of large-scale systems," Journal of Process Control, vol. 21, no. 5, pp. 775–786, June 2011.

[14] F. Tedesco, C. Ocampo-Martinez, A. Casavola, and V. Puig, "Centralized and distributed command governor approaches for water supply systems management," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 4, pp. 586–595, April 2018.

[15] A. Rahman and Z. Y. Wu, "Multistep simulation-optimization modeling approach for partitioning water distribution system into district meter areas," Journal of Water Resources Planning and Management, vol. 144, no. 5, May 2018.

[16] J. Li, C.-C. Liu, and K. P. Schneider, "Controlled partitioning of a power network considering real and reactive power balance," IEEE Transactions on Smart Grid, vol. 1, no. 3, pp. 261–269, December 2010.

[17] A. Golshani, W. Sun, and K. Sun, "Advanced power system partitioning method for fast and reliable restoration: toward a self-healing power grid," IET Generation, Transmission & Distribution, vol. 12, no. 1, pp. 42–52, January 2018.

[18] J. Anderson, Y. Chang, and A. Papachristodoulou, "Model decomposition and reduction tools for large-scale networks in systems biology," Automatica, vol. 47, no. 6, pp. 1165–1174, June 2011.

[19] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, VLSI Physical Design: From Graph Partitioning to Timing Closure. Springer, 2011.

[20] Z. Zhou, S. Lin, and Y. Xi, "A dynamic network partition method for heterogenous urban traffic networks," in Proceedings of the 15^{th} International IEEE Conference on Intelligent Transportation Systems, Anchorage, Alaska, USA, September 2012, pp. 820–825.

[21] F. J. Muros, J. M. Maestre, E. Algaba, C. Ocampo-Martinez, and E. F. Camacho, "An application of the Shapley value to perform system partitioning," in Proceedings of the 33^{rd} American Control Conference (ACC 2015), Chicago, IL, USA, July 2015, pp. 2143–2148.

[22] F. J. Muros, J. M. Maestre, E. Algaba, T. Alamo, and E. F. Camacho, "Networked control design for coalitional schemes using game-theoretic methods," Automatica, vol. 78, pp. 320–332, April 2017.

[23] F. J. Muros, J. M. Maestre, C. Ocampo-Martinez, E. Algaba, and E. F. Camacho, "Partitioning of large-scale systems using game-theoretic coalitional methods," in Proceedings of the 16^{th} European Control Conference (ECC 2018), Limassol, Cyprus, June 2018, pp. 2517–2522.

[24] F. Fele, J. M. Maestre, and E. F. Camacho, "Coalitional control: Cooperative game theory and control," IEEE Control Systems Magazine, vol. 37, no. 1, pp. 53–69, February 2017.

[25] D. Bauso and G. Notarstefano, "Distributed n-player approachability and consensus in coalitional games," IEEE Transactions on Automatic Control, vol. 60, no. 11, pp. 3107–3112, November 2015.

[26] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Transactions on Automatic Control, vol. 49, no. 9, pp. 1520–1533, September 2004.

[27] L. S. Shapley, "A value for n-person games," in Contributions to the Theory of Games II. Annals of Mathematics Studies, vol. 28, H. W. Kuhn and A. W. Tucker, Eds. Princeton, New Jersey, USA: Princeton University Press, 1953, pp. 307–317.

[28] J. Castro, D. Gómez, and J. Tejada, "Polynomial calculation of the Shapley value based on sampling," Computers & Operations Research, vol. 36, no. 5, pp. 1726–1730, May 2009.

[29] H. Ishii and R. Tempo, "Distributed randomized algorithms for the PageRank computation," IEEE Transactions on Automatic Control, vol. 55, no. 9, pp. 1987–2002, February 2010.

[30] R. A. DeVore and V. N. Temlyakov, "Some remarks on greedy algorithms," Advances in Computational Mathematics, vol. 5, no. 1, pp. 173–187, December 1996.

[31] J. Qin, Y. Chow, J. Yang, and R. Rajagopal, "Distributed online modified greedy algorithm for networked storage operation under uncertainty," IEEE Transactions on Smart Grid, vol. 7, no. 2, pp. 1106–1118, March 2016.

[32] J. M. Maestre and H. Ishii, "A PageRank based coalitional control scheme," International Journal of Control, Systems and Automation. In press, 2017.

[33] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," IEEE Control Systems, vol. 34, no. 4, pp. 87–97, August 2014.

[34] R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," Annals of the History of Computing, vol. 7, no. 1, pp. 43–57, January-March 1985.

[35] J. Li, C.-C. Liu, and K. P. Schneider, "Distribution system restoration with microgrids using spanning tree search," IEEE Transactions on Power Systems, vol. 29, no. 6, pp. 3021–3029, November 2014.

[36] H. W. Becker and J. Riordan, "The arithmetic of Bell and Stirling numbers," American Journal of Mathematics, vol. 70, no. 2, pp. 385–394, April 1948.

[37] M. Brdys and B. Ulanicki, Operational Control of Water Systems: Structures Algorithms and Applications. UK: Prentice Hall International, 1994.

[38] A. K. Sampathirao, P. Sopasakis, A. Bemporad, and P. P. Patrinos, "GPU-accelerated stochastic predictive control of drinking water networks," IEEE Transactions on Control Systems Technology, vol. 26, no. 2, pp. 551–562, March 2018.

[39] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," Proceedings of the National Academy of Sciences of the United States of America, vol. 99, no. 12, pp. 7281–7826, June 2002.

[40] U. Brandes, "On variants of shortest-path betweenness centrality and their generic computation," Social Networks, vol. 30, no. 2, pp. 136–145, May 2008.

FRANCISCO JAVIER MUROS received the M.S. degree in Telecommunications Engineering from the University of Seville in 2004, and the Ph.D. on automation, robotics and telematics, summa cum laude, by the same university in 2017. Since 2005, he has acquired a wide experience in several works related to the electricity sector. He received a Master's degree in Design, Construction and Maintenance of High Voltage Electrical Transmission Infrastructure from the Comillas Pontifical University, Madrid in 2014, being between 2015 and 2016 Coordinator of this Master. He performed a Ph.D.-stay in Maastricht, The Netherlands, in 2016. At present he is with the Department of Systems and Automation Engineering at the University of Seville. He is author and co-author of more than twenty publications including journal papers, book chapters, and conference papers. He has participated in the European Union Project DYMASOS and in several MINECO-Spain Projects. His research interests focus on cooperative and noncooperative game theory, coalitional and distributed control.

JOSÉ MARÍA MAESTRE (M'14–SM'18) received the joint B.Eng. and M.Eng. degree in telecommunication engineering from the University of Seville in 2005, and the Ph.D. on automation and robotics in 2010 by the same university, where he also works as associate professor. He has also worked in LTH at Lund University (guest researcher), TU Delft (postdoc), University of Cadiz (assistant professor), and Tokyo Institute of Technology (visiting professor). Besides his Ph.D., which was awarded with the extraordinary prize of the University of Seville, he has a master degree in smart homes and intelligent buildings from the Universidad Politécnica de Madrid, a master in telecommunications economics from the Universidad Nacional de Educación a Distancia, and a master in economics and development from the University of Seville. His main research interests are the control of distributed systems and the integration of service robots in the smart home. He has authored and coauthored more than one hundred conference and journal papers regarding these topics. He is also editor of the books *Service Robotics within the Digital Home: Applications and Future Prospects* (Springer, 2011), *Distributed Model Predictive Control Made Easy* (Springer, 2014), and *Domótica para Ingenieros* (Paraninfo, 2015). Finally, he is one of the founders of the technological firms *Idener* and *Eskesso*.

CARLOS OCAMPO-MARTINEZ (M'11–SM'13) received his Electronics Engineering degree and MSc. degree in Industrial Automation from the National University of Colombia, Campus Manizales in 2001 and 2003, respectively. In 2007, he received the Ph.D. degree in Control Engineering from the Technical University of Catalonia - BarcelonaTech (Barcelona, Spain). Since 2011, he is with the Universitat Politècnica de Catalunya, Automatic Control Department (ESAII), currently as Associate Professor in automatic control and model predictive control. Between 2014 and 2018, he was also Deputy Director of the Institut de Robòtica i Informàtica Industrial. His main research interests include constrained model predictive control, large-scale systems management (partitioning and noncentralized control), and industrial applications (mainly related to the key scopes of water and energy).

ENCARNACIÓN ALGABA is Associate Professor in the Department of Applied Mathematics II of the Advanced Engineering School at the University of Seville and member of the Mathematics Research Institute at the University of Seville (IMUS). She received her Ph.D. degree in Mathematical Sciences from the University of Seville. Her main research line is about Cooperative Games on Combinatorial Structures with relevant contributions in the area. She is one of the Spanish representatives in the SING (Spain, Italy, Netherlands Game Theory) Committee from 2012 and member of the Scientific Committee in SING 10 (Cracow, Poland 2014), European meeting on Game Theory SING 11-GTM 2015 (St. Petersburg, Russia 2015), SING 12 (Odense, Denmark 2016), SING 13 (Paris, France 2017) and SING 14 (Bayreuth, Germany 2018). She has acted as evaluator of projects at national and European level and she belongs to the national expert evaluators board from 2014. She has organized diverse courses, streams in International Conferences and workshops on Game Theory. She has been invited speaker and guest researcher in various universities, and has given invited Summer Courses and numerous talks in International Conferences.

EDUARDO F. CAMACHO (M'84–SM'00–F'11) received the Ph.D. degree in electrical engineering from the University of Seville, Seville, Spain.

He is now a Full Professor with the Department of System and Automation Engineering, University of Seville. He is author of *Model Predictive Control in the Process Industry* (1995), *Advanced Control of Solar Plants* (1997), *Model Predictive Control* (1999), (Springer, 2004, 2^{nd} edition), *Control e Instrumentación de Procesos Químicos* (Ed. Síntesis), *Control of Dead-time Processes* (Springer, 2007) and *Control of Solar Systems* (Springer, 2011). He has served on various IFAC technical committees and chaired the IFAC publication Committee from 2002-2005. He was the president of the European Control Association (2005-2007) and chaired the IEEE/CSS International Affairs Committee (2003-2006), Chair of the IFAC Policy Committee and a member of the IEEE/CSS Board of Governors. He has acted as evaluator of projects at national and European level and was appointed Manager of the Advanced Production Technology Program of the Spanish National R&D Program (1996-2000). He was one of the Spanish representatives on the Program Committee of the Growth Research program and expert for the Program Committee of the NMP research priority of the European Union. He has carried out reviews and editorial work for various technical journals and many conferences. At present he is one of the Editors of the *IFAC Journal, Control Engineering Practice*, Editor-at-Large of the *European Journal of Control* and Subject Editor of *Optimal Control: Applications and Methods*.

Dr. Camacho is an IEEE and IFAC Fellow. He was Publication Chair for the IFAC World Congress 2002, General Chair of the joint 44^{th} IEEE CDC-ECC 2005, and Co-General Chair of the joint 50^{th} IEEE CDC-ECC 2011.