

# Time-varying Partitioning for Predictive Control Design: Density-Games Approach

Julian Barreiro-Gomez <sup>a</sup>, Carlos Ocampo-Martinez <sup>b</sup>, and Nicanor Quijano <sup>c</sup>

<sup>a</sup>*Learning & Game Theory Laboratory, Engineering Division, New York University in Abu Dhabi (NYUAD), Saadiyat Campus PO Box 129188, Abu Dhabi, UAE*

<sup>b</sup>*Automatic Control Department, Universitat Politècnica de Catalunya, Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas, 4-6, 08028 Barcelona, Spain*

<sup>c</sup>*Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Carrera 1 No 18A-10, Bogotá Colombia*

---

## Abstract

The design of distributed optimization-based controllers for large-scale systems (LSSs) implies every time new challenges. The fact that LSSs are generally located throughout large geographical areas makes difficult the recollection of measurements and their transmission. In this regard, the communication network that is required for a centralized control approach might have high associated economic costs. Furthermore, the computation of a large amount of data implies a high computational burden to manage, process and use them in order to make decisions over the system operation. A plausible solution to mitigate the aforementioned issues associated with the control of LSSs consists in dividing this type of systems into smaller sub-systems able to be handled by independent local controllers. This paper studies two fundamental components of the design of distributed optimization-based controllers for LSSs, i.e., the system partitioning and distributed optimization algorithms. The design of distributed model predictive control (DMPC) strategies with a system partitioning and by using density-dependent population games (DDPG) is presented.

*Key words:* Predictive control, system partitioning, density games, population dynamics, distributed control, plug-and-play features.

---

## 1 Introduction

The increasing emergence of large-scale systems (LSSs), e.g., water distribution systems, smart grids, or traffic systems, have promoted the study of model predictive control (MPC) under non-centralized schemes. This fact is also motivated by two different aspects, i.e., communication issues (infrastructure) to collect and transmit data associated with the system states, and computational issues to determine the appropriate control inputs. Hence, the problem of obtaining non-centralized control formulations has become a relevant research topic. The process of making controllers non-centralized is normally addressed by dividing the whole system into  $m$  different sub-systems and by designing several local and smaller controllers. In [1–4], and more recently in [5], a wide discussion related to the design of non-centralized MPC controllers is developed. Furthermore, there are several classifications within the non-centralized MPC

controllers depending on their architecture and on how different local controllers share information to one another [4]. One of the non-centralized configurations corresponds to decentralized MPC controllers, where sub-systems might have a dynamical coupling among them. In the decentralized MPC architecture [6], there is a set of local MPC controllers (each one in charge of the control of a sub-system), which do not exchange information to one another. Therefore, in order to implement this control architecture, it is usually assumed that the dynamical coupling among sub-systems is weak, for which these non-centralized configurations have some limitations as studied in [7]. Therefore, the performance of the closed-loop system can be enhanced by considering that local MPC controllers can exchange information. Then, these local controllers should be coordinated to obtain a control input [8]. This modification adding available information among controllers leads to the distributed MPC (DMPC) architecture. There are two main aspects to design distributed optimization-based controllers, i.e., (i) the identification of the sub-systems composing the whole system that is desired to be controlled, and (ii) the appropriate coordination

---

*Email addresses:* jbarreiro@nyu.edu (Julian Barreiro-Gomez), cocampo@iri.upc.edu (Carlos Ocampo-Martinez), nquijano@uniandes.edu.co (Nicanor Quijano).

among the different local controllers associated with subsystems. The former component consists in a partitioning problem making considerations related to dynamical coupling, existing coupled physical and/or operational constraints, and communication requirements; whereas the latter component requires the development of distributed optimization strategies given a communication structure.

This paper addresses both components involved into the design of distributed optimization-based controllers, i.e., system partitioning and distributed optimization. Regarding the former component, the system partitioning problem has gotten increasing importance in the automatic control community as systems become larger and more challenging, and as the requirements and desired closed-loop performance become more strict. Many partitioning proposals focus on specific dynamical systems, e.g., in thermal control [9], control of electric power distribution systems [10,11], or on a particular control strategy, e.g., in decomposition structure of distributed predictive controllers [12,13]. Regarding the second component for distributed controllers, this paper discusses the distributed optimization based on dynamic games, which have become a quite useful tool in the design of distributed controllers as it has been presented in [14,15]. This paper focuses on the density-dependent games where variations of the population size is allowed, illustrating a situation in which death and birth, or reproduction rates, are considered as in [16,17]. These dynamics have not been either deduced from a version of the general dynamics known as *mean dynamics* (Kolmogorov forward equation), and by imposing different rules on revision protocols, nor proposed in a distributed information-sharing fashion.

The contribution of this paper is threefold, i.e.,

- (1) First, a novel partitioning approach based on a non-directed graph representing information sharing inspired by the Kernighan-Lin algorithm [18] is presented.
- (2) Secondly, it is proposed to extend the mean dynamics, which are used in the deduction of population dynamics [19], considering strategy-interaction constraints and a reproduction-rate parameter, i.e., the density-dependent mean dynamics with non-complete population-interaction structures. Then, the distributed density-dependent replicator, Smith, and projection dynamics are deduced. Afterwards, it is shown that these density dynamics may be used to solve distributed constrained optimization problems.
- (3) As a third contribution, and taking advantage of the properties that density games have, a DMPC controller design is proposed based on the distributed DDPG, i.e., an algorithm for dynamical partitioning a communication structure and develop a distributed network control scheme is provided. This

is made by means of the combination of the two previous mentioned contributions.

It is shown that the population-interaction structure can be modified dynamically along the time by adding conditions over the optimization problem constraints depending on the current system state, leading this fact to a time-varying information-sharing network for control purposes.

The remainder of this paper is organized as follows. Section 2 presents the addressed problem statement showing a general scheme and the optimization problem corresponding to the distributed LSSs. Therefore, Sections 3 and 4 present the contribution of the paper, i.e., the distributed system partitioning algorithm, the distributed density-dependent population games with their features, and how a distributed constrained optimization problem can be solved by using the proposed game-theoretical approach. Section 5 introduces different control approaches that can be designed by using the proposed system partitioning and distributed density games. Then, in Section 6, a case study is introduced, the proposed approach is implemented and results are discussed. Finally, conclusions are drawn in Section 7.

*Notation:* Column vectors are denoted by bold style, e.g.,  $\mathbf{p}$ . Matrices are denoted by bold upper case, e.g.,  $\mathbf{A}$ . Differently, scalars are denoted by non-bold style, e.g.,  $n$ . The sets are denoted by calligraphic upper case, e.g.,  $\mathcal{S}$ . The norm  $\|\mathbf{x}\|_{\mathbf{Q}}$  is defined as  $\|\mathbf{x}\|_{\mathbf{Q}} = \mathbf{x}^T \mathbf{Q} \mathbf{x}$ . The function  $[\cdot]_+ = \max(0, \cdot)$  is used to simplify the notation. The identity matrix of size  $n \times n$  is denoted by  $\mathbf{I}_n$ ,  $\mathbf{1}_n$  is the column vector with  $n$  unitary entries, i.e.,  $\mathbf{1}_n = [1 \ \dots \ 1]^T \in \mathbb{R}^n$ , similarly  $\mathbf{0}_n$  is the column vector with  $n$  null entries, i.e.,  $\mathbf{0}_n = [0 \ \dots \ 0]^T \in \mathbb{R}^n$ , in addition,  $\mathbf{0}_{n \times l}$  is the matrix of null entries and dimension  $n \times l$ , and  $\text{diag}(\mathbf{x})$  is the diagonal matrix of the vector  $\mathbf{x}$ . Let  $\mathbf{A} = [a_{ij}]$  and  $\mathbf{B} = [b_{ij}]$  be matrices with the same dimension, i.e.,  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times l}$ , then let  $\mathbf{C} = [\mathbf{A} \circ \mathbf{B}] \in \mathbb{R}^{n \times l}$  be the Hadamard product, i.e.,  $[c_{ij}] = [a_{ij}][b_{ij}]$ , for all  $i = \{1, \dots, n\}$  and  $j = \{1, \dots, l\}$ . If matrix  $\mathbf{A}$  is positive (negative) semi-definite, it is denoted by  $\mathbf{A} \geq 0$  ( $\mathbf{A} \leq 0$ ). Throughout this paper, both continuous-time and discrete-time systems are treated. Continuous time is denoted by  $t$  and it is mostly omitted throughout the paper in order to simplify the notation. Moreover,  $\dot{x}$  denotes the derivative with respect to continuous time, i.e.,  $\dot{x} = \frac{d}{dt}x(t)$ . In contrast,  $k \in \mathbb{Z}_{\geq 0}$  denotes the discrete time.

## 2 Problem Statement

Consider a state-space discrete-time system with a sampling time  $\Delta t$  and represented by the following model:

$$\mathbf{x}_{k+1} = \mathbf{A}_d \mathbf{x}_k + \mathbf{B} \mathbf{u}_k + \mathbf{B}_d \mathbf{d}_k, \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^{n_x}$  is the system state vector,  $\mathbf{u} \in \mathbb{R}^{n_u}$  is the vector of control inputs,  $\mathbf{d} \in \mathbb{R}^{n_d}$  denotes the vector of disturbances that affect the system, and the state-space matrices are given by  $\mathbf{A}_d \in \mathbb{R}^{n_x \times n_x}$ ,  $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ , and  $\mathbf{B}_d \in \mathbb{R}^{n_x \times n_d}$ . The states and control inputs are subject to physical and operational constraints, which define feasible sets denoted by  $\mathcal{X} \triangleq \{\mathbf{x} \in \mathbb{R}^{n_x} : \mathbf{x}^{\min} \leq \mathbf{x} \leq \mathbf{x}^{\max}\}$ , and  $\mathcal{U} \triangleq \{\mathbf{u} \in \mathbb{R}^{n_u} : \mathbf{u}^{\min} \leq \mathbf{u} \leq \mathbf{u}^{\max}\}$ , where vectors  $\mathbf{x}^{\min}$  and  $\mathbf{x}^{\max}$  correspond to the lower and upper limits for the system states, respectively. Similarly, vectors  $\mathbf{u}^{\min}$  and  $\mathbf{u}^{\max}$  denote the lower and upper limits for the control inputs, respectively. The control sequence for a fixed-time prediction horizon, represented by  $N \in \mathbb{Z}_{>0}$  at the instant time  $k \in \mathbb{Z}_{\geq 0}$ , is denoted by  $\hat{\mathbf{u}}_k$ . When the control input sequence  $\hat{\mathbf{u}}_k$  is applied to the system (1) with initial state  $\mathbf{x}_{k|k} \triangleq \mathbf{x}_k$ , a system states sequence  $\hat{\mathbf{x}}_k$  is generated. Finally, the disturbances time-varying sequence along the horizon  $N$  is denoted by  $\hat{\mathbf{d}}_k$  and it is assumed to be bounded and known from a forecasting algorithm [20,21], i.e.,

$$\begin{cases} \hat{\mathbf{u}}_k \triangleq (\mathbf{u}_{k|k}, \mathbf{u}_{k+1|k}, \dots, \mathbf{u}_{k+N-1|k}), \\ \hat{\mathbf{x}}_k \triangleq (\mathbf{x}_{k+1|k}, \mathbf{x}_{k+2|k}, \dots, \mathbf{x}_{k+N|k}), \\ \hat{\mathbf{d}}_k \triangleq (\mathbf{d}_{k|k}, \mathbf{d}_{k+1|k}, \dots, \mathbf{d}_{k+N-1|k}). \end{cases} \quad (2)$$

The system with discrete-time model as in (1) is controlled with an MPC controller whose optimization problem  $\mathbf{P}_{\text{MPC}}$  is stated as follows:

$$\min_{\hat{\mathbf{u}}_k} J(\mathbf{x}_k, \mathbf{u}), \quad (3a)$$

s. t.

$$\mathbf{x}_{k+j+1|k} = \mathbf{A}_d \mathbf{x}_{k+j|k} + \mathbf{B} \mathbf{u}_{k+j|k} + \mathbf{B}_d \mathbf{d}_{k+j|k}, \quad (3b)$$

$$\mathbf{u}_{k+j|k} \in \mathcal{U}, \quad (3c)$$

$$\mathbf{x}_{k+j|k} \in \mathcal{X}, \quad (3d)$$

$$\mathbf{E} \begin{bmatrix} \mathbf{u}_{k+j|k}^\top & \mathbf{d}_{k+j|k}^\top \end{bmatrix}^\top \in \mathcal{D}, \quad (3e)$$

where (3b), (3c) and (3e), for all  $j \in [0, N-1] \cap \mathbb{Z}_{\geq 0}$ , and (3d), for all  $j \in [1, N] \cap \mathbb{Z}_{\geq 0}$ . Hence,  $\mathbf{E} \in \mathbb{R}^{r \times (n_u + n_d)}$  is associated to  $r$  mass balance terms between control inputs and disturbances constrained by the feasible set  $\mathcal{D}$ . Assuming that  $\mathbf{P}_{\text{MPC}}$  is feasible, an optimal sequence is computed, and only the first element of this sequence is applied to the system, obtaining the corresponding evolution of the system. Then, following the MPC philosophy, a new optimization problem is formulated for the next time instant [22].

This paper focuses on making the aforementioned MPC controller perform in a distributed fashion by means of both a partitioning algorithm, and distributed optimization algorithms based on evolutionary density games. Once these two tools are presented (partitioning

and optimization algorithms), they are used to design a distributed MPC controller, recalling the optimization problem in (3). Next, both the partitioning algorithm and the evolutionary-games-based optimization algorithm are presented.

### 3 Distributed LSS Partitioning Algorithm

Consider an information-sharing network whose topology is represented by an undirected connected graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ , where  $\mathcal{S} = \{1, \dots, n\}$  represents the set of  $n \in \mathbb{Z}_{>1}$  nodes associated to the control-strategy variables, and  $\mathcal{E} \subset \{(i, j) : i, j \in \mathcal{S}\}$  denoted the set of edges of  $\mathcal{G}$  representing the possible information sharing among nodes  $\mathcal{S}$ . Notice that the graph is undirected assuming that the edges represent bidirectional-information channels. Hence,  $\mathbf{A} \in \{0, 1\}^{n \times n}$  denotes the adjacency matrix whose elements  $a_{ij} = 1$  if  $(i, j) \in \mathcal{E}$ , and  $a_{ij} = 0$ , otherwise. This network is determined by taking into account all the required information in order to compute the optimal control inputs. Finally, the set of neighbors for each node  $i \in \mathcal{S}$  is defined as  $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}\}$ .

Some parameters associated to the characteristics of the information-sharing graph  $\mathcal{G}$  are defined. First, a function defining the distance among different nodes from the set  $\mathcal{S}$  is defined as  $d : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ . Moreover, there is a constant matrix  $\mathbf{D} \in \mathbb{R}_{\geq 0}^{n \times n}$  summarizing all the evaluated distances for all the different nodes from  $\mathcal{S}$ , and whose elements are given by  $d_{ij} = d(i, j)$ . As a second parameter, let  $c(i, j)$  be a function that determines a relevance factor, i.e., how relevant the information shared between node  $j \in \mathcal{S}$  and node  $i \in \mathcal{S}$  is. Thus, there is a constant matrix  $\mathbf{C} \in \mathbb{R}_{\geq 0}^{n \times n}$  representing all the relevance factors as  $c_{ij} = c(i, j)$ . On the other hand, consider a set of indexes denoted by  $\mathcal{K} = \{1, \dots, m\}$  where  $m \in \mathbb{Z}_{>0}$  indicates the number of partitions of the graph  $\mathcal{G}$ . Therefore, the obtained partitioning at a time instant  $k$  is denoted by the set  $\mathcal{P}_k = \{\mathcal{S}_k^\ell : \ell \in \mathcal{K}\}$ . Each partitioning of  $\mathcal{G}$  at time instant  $k$  is an undirected connected graph  $\mathcal{G}_k^\ell = (\mathcal{S}_k^\ell, \mathcal{E}_k^\ell)$ , for all  $\ell \in \mathcal{K}$ , where  $\bigcap_{\ell \in \mathcal{K}} \mathcal{S}_k^\ell = \emptyset$ , and  $\bigcup_{\ell \in \mathcal{K}} \mathcal{S}_k^\ell = \mathcal{S}$ , for all  $k$ . Hence, for a given partitioning  $\mathcal{P}_k$ , there is a function  $g : \mathcal{S} \rightarrow \mathcal{K}$  receiving a node  $i \in \mathcal{S}$  and returning an index  $\ell \in \mathcal{K}$  corresponding to the partition to which the node  $i \in \mathcal{S}$  belongs to, i.e.,  $g(i) = \{\ell \in \mathcal{K} : i \in \mathcal{S}_k^\ell\}$ . Finally, there is a parameter allowing to compare two neighbor partitions in  $\mathcal{G}$ . Consider the time-varying matrix  $\mathbf{V}_k \in \mathbb{R}^{n \times n}$  whose elements  $v_{ij,k} = |\mathcal{S}_k^{g(i)}| - |\mathcal{S}_k^{g(j)}|$  if  $g(i) \neq g(j)$ , being nodes  $i$ , and  $j$  neighbors, i.e.,  $j \in \mathcal{N}_i$ .

**Remark 1** Let  $\mathcal{P}_k$  be an admissible partition, i.e.,  $\bigcap_{\ell \in \mathcal{K}} \mathcal{S}_k^\ell = \emptyset$ , and  $\bigcup_{\ell \in \mathcal{K}} \mathcal{S}_k^\ell = \mathcal{S}$ . Therefore,  $g : \mathcal{S} \rightarrow \mathcal{K}$  is a function since the set  $\{\ell \in \mathcal{K} : i \in \mathcal{S}_k^\ell\}$  is a singleton, for all  $\ell \in \mathcal{K}$ , and  $g(i)$  is a function, for all  $i \in \mathcal{S}$ . ■

#### 3.1 Partitioning Problem Statement

In order to evaluate a partition of  $\mathcal{G}$  at a time in-

stant  $k$ , which is denoted by  $\mathcal{P}_k$ , four indicators  $\sigma_1(\mathcal{P}_k), \dots, \sigma_4(\mathcal{P}_k)$  corresponding to links, nodes, distance and relevance, are defined as follows:

*Links:* Indicates the amount of links connecting different partitions, i.e., links  $(i, j) \in \mathcal{E}$  such that  $(i, j) \notin \mathcal{E}_k^\ell$ , for all  $\ell \in \mathcal{K}$ ,

$$\sigma_1(\mathcal{P}_k) = \frac{1}{2} \sum_{\ell \in \mathcal{K}} \sum_{i \in \mathcal{S}_k^\ell} \sum_{j \in \mathcal{S} \setminus \mathcal{S}_k^\ell} a_{ij}.$$

*Nodes:* Indicates the difference between the amount of nodes of partitions given by  $|\mathcal{S}_k^\ell|$  for neighbor partitions along the time. This indicator is expressed in terms of the time-varying matrix  $\mathbf{V}_k$ , i.e.,

$$\sigma_2(\mathcal{P}_k) = \sum_{\ell \in \mathcal{K}} \sum_{i \in \mathcal{S}_k^\ell} \sum_{j \in \mathcal{S} \setminus \mathcal{S}_k^\ell} v_{ij,k}.$$

*Distance:* Indicates the inverse of the distance among partitions. This is expressed in terms of the inverse of the distance of the links connecting different partitions. Notice that the inverse is considered just to make the indicator bigger as two partitions are close. Then

$$\sigma_3(\mathcal{P}_k) = \frac{1}{2} \sum_{\ell \in \mathcal{K}} \sum_{i \in \mathcal{S}_k^\ell} \sum_{j \in \mathcal{S} \setminus \mathcal{S}_k^\ell} d_{ij}^{-1}.$$

*Relevance:* Indicates the relevance of the information that is being shared among different partitions. Notice that this indicator also gives a notion about the impact when having information loss due to the isolation of partitions, i.e., if the indicator is low, then there is not a big impact when disconnected the communication among partitions. This indicator is given by

$$\sigma_4(\mathcal{P}_k) = \sum_{\ell \in \mathcal{K}} \sum_{i \in \mathcal{S}_k^\ell} \sum_{j \in \mathcal{S} \setminus \mathcal{S}_k^\ell} c_{ij},$$

and can be tuned by taking into account the required information to satisfy constraints. Thus, the partitioning algorithm will seek an appropriate way to partition the system into sub-systems that can operate in a decoupled manner. The  $m$ -partitioning problem consists in computing the optimal partition set  $\mathcal{P}^*$  such that the indicators are minimized. Therefore, the optimal partitioning  $\mathcal{P}^*$  is obtained by solving the following optimization problem:

$$\min_{\mathcal{P}} \sum_{j=1}^4 \varphi_j \sigma_j(\mathcal{P}_k), \quad (4a)$$

$$\text{s. t. } \bigcap_{\ell \in \mathcal{K}} \mathcal{S}_k^\ell = \emptyset, \quad (4b)$$

$$\bigcup_{\ell \in \mathcal{K}} \mathcal{S}_k^\ell = \mathcal{S}. \quad (4c)$$

### 3.2 Distributed Partitioning Algorithm

Consider the weighted graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathbf{W}_k)$ , where  $\mathbf{W}_k \in \mathbb{R}^{n \times n}$  is a time-varying weighting matrix. The elements of the matrix  $\mathbf{W}_k$ , denoted by  $w_{ij,k} \in \mathbb{R}_{\geq 0}$ , represent a cost associated to the each link  $(i, j) \in \mathcal{E}$ , where  $w_{ij,k} = \varphi_1 a_{ij} + \varphi_2 v_{ij,k} + \varphi_3 d_{ij}^{-1} + \varphi_4 c_{ij}$ . Hence, in order to solve the optimization problem (4), it is proposed to solve the  $m$ -partitioning problem (4) as follows:

$$\min_{\mathcal{P}} \sum_{\ell \in \mathcal{K}} \sum_{i \in \mathcal{S}_k^\ell} \sum_{j \in \mathcal{S} \setminus \mathcal{S}_k^\ell} w_{ij,k}, \quad (5)$$

and subject to constraints (4b) and (4c). The set of nodes  $\mathcal{S}_k^\ell$  of the subgraph  $\mathcal{G}_k^\ell$  is composed of a set of internal nodes denoted by  $\tilde{\mathcal{S}}_k^\ell$ , and a set of external nodes denoted by  $\hat{\mathcal{S}}_k^\ell$ , for all  $\ell \in \mathcal{K}$ . The internal nodes from the set  $\tilde{\mathcal{S}}_k^\ell$  only have connection to nodes that belong to the same partition. In contrast, the external nodes from the set  $\hat{\mathcal{S}}_k^\ell$  have connection to at least one node that belongs to a different partition. Formally,

$$\tilde{\mathcal{S}}_k^\ell = \{i \in \mathcal{S}_k^\ell : \mathcal{N}_i \subseteq \mathcal{S}_k^\ell\}, \quad \forall \ell \in \mathcal{K}, \quad (6a)$$

$$\hat{\mathcal{S}}_k^\ell = \{i \in \mathcal{S}_k^\ell : \mathcal{N}_i \not\subseteq \mathcal{S}_k^\ell\}, \quad \forall \ell \in \mathcal{K}, \quad (6b)$$

$$\mathcal{S}_k^\ell = \tilde{\mathcal{S}}_k^\ell \cup \hat{\mathcal{S}}_k^\ell, \quad \forall \ell \in \mathcal{K}.$$

Each external node  $i \in \hat{\mathcal{S}}_k^\ell$ , for all  $\ell \in \mathcal{K}$ , represents a decision maker that is able to select a partition from the set

$$\mathcal{Q}_{i,k} = \{g(j) : j \in \mathcal{N}_i\} \setminus \{g(i)\}. \quad (7)$$

Moreover, each decision maker  $i \in \{\cup_{\ell \in \mathcal{K}} \hat{\mathcal{S}}_k^\ell\}$ , at time instant  $k$ , has associated an internal cost denoted by  $\check{h}_i$ , i.e.,

$$\check{h}_i(\mathcal{P}_k) = \sum_{j \in \mathcal{S}_k^{g(i)} \cap \mathcal{N}_i} w_{ij,k}, \quad \forall i \in \hat{\mathcal{S}}_k^\ell, \text{ and } \ell \in \mathcal{K},$$

and an external benefit denoted by  $\hat{h}_i^\ell$ , for each partition  $\ell$  from the set  $\mathcal{Q}_{i,k}$ , i.e.,

$$\hat{h}_i^\ell(\mathcal{P}_k) = \sum_{j \in \mathcal{S}_k^\ell \cap \mathcal{N}_i} w_{ij,k}, \quad \forall i \in \hat{\mathcal{S}}_k^\ell, \text{ and } \ell \in \mathcal{Q}_{i,k}.$$

Then, the best external benefit for the decision maker  $i \in \hat{\mathcal{S}}_k^\ell$ , for all  $\ell \in \mathcal{Q}_{i,k}$ , is obtained as follows:

$$\hat{h}_i(\mathcal{P}_k) = \max_{\ell \in \mathcal{Q}_{i,k}} \hat{h}_i^\ell(\mathcal{P}_k).$$

Finally, as previously mentioned, the decision maker selects among the possible available partitions depending on a utility denoted by  $\eta_i$ , i.e., if the decision maker has

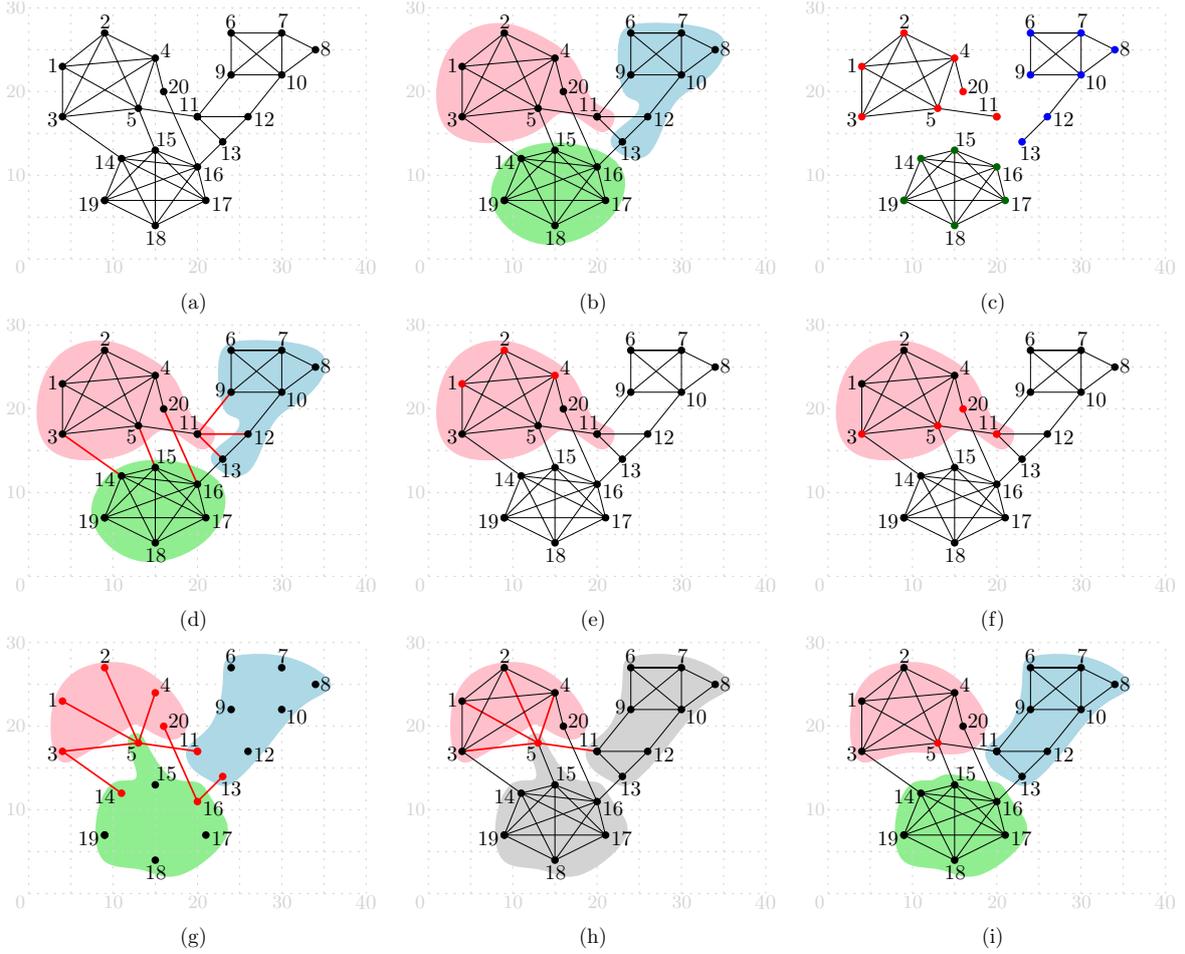


Fig. 1. Illustrative notation example corresponding to the partitioning algorithm.

incentives to move from one partition to another one, then

$$\eta_i(\mathcal{P}_k) = \max(0, \hat{h}_i(\mathcal{P}_k) - \check{h}_i(\mathcal{P}_k)), \quad \forall i \in \hat{\mathcal{S}}_k^\ell, \text{ and } \ell \in \mathcal{K}.$$

Further, consider an undirected graph  $\tilde{\mathcal{G}}_k = (\tilde{\mathcal{S}}_k, \tilde{\mathcal{E}}_k)$  at time instant  $k$  –not necessarily connected– composed of all the decision makers –external nodes of  $\mathcal{G}$ –. Then, the set of nodes of  $\tilde{\mathcal{G}}_k$  is given by  $\tilde{\mathcal{S}}_k = \bigcup_{\ell \in \mathcal{K}} \hat{\mathcal{S}}_k^\ell$ , and the set of links is given by  $\tilde{\mathcal{E}}_k \subset \{(i, j) : i \in \hat{\mathcal{S}}_k^\ell, j \in \hat{\mathcal{S}}_k^r, \ell \neq r\}$ .

Let  $\tilde{\mathbf{A}}_k \in \{0, 1\}^{|\tilde{\mathcal{S}}_k| \times |\tilde{\mathcal{S}}_k|}$  be the adjacency matrix of the graph. Since  $\tilde{\mathcal{G}}_k$  is not necessarily a connected graph, it has  $q \in \mathbb{Z}_{>0}$  components at time instant  $k$ , where the set of components of the graph is  $\mathcal{C}_k = \{1, \dots, q\}$ . Each component is a graph denoted by  $\tilde{\mathcal{G}}_k^z = (\tilde{\mathcal{S}}_k^z, \tilde{\mathcal{E}}_k^z)$ , with adjacency matrix  $\tilde{\mathbf{A}}_k^z$ , is connected and not necessarily complete, for all  $z \in \mathcal{C}_k$ . Furthermore, only the decision maker with higher incentives (winner decision maker in the component  $z \in \mathcal{C}_k$  denoted by  $i_k^{z*} \in \mathcal{W}_k^z \subseteq \tilde{\mathcal{S}}_k^z$ ) would make a decision to switch from its current partition to

another one among the set of available partitions, i.e.,

$$i_k^{z*} \in \arg \max_{i \in \tilde{\mathcal{S}}_k^z} \eta_i(\mathcal{P}_k) = \mathcal{W}_k^z, \quad \forall z \in \mathcal{C}_k. \quad (8)$$

Notice that (8) should be solved at each time instant  $k \in \mathbb{Z}_{\geq 0}$ . The best option for the decision maker  $i_k^{z*} \in \tilde{\mathcal{S}}_k^z$ , for all  $z \in \mathcal{C}_k$ , to select a new partition is

$$\ell_k^{z*} \in \arg \max_{\ell \in \mathcal{Q}_{i_k^{z*}, k}} \hat{h}_{i_k^{z*}}^\ell(\mathcal{P}_k). \quad (9)$$

Hence, the partitioning is modified only if  $\eta_i(\mathcal{P}_k) > \kappa$ , where  $\kappa \in \mathbb{R}_{>0}$  establishes an ending-up condition. The updating is performed as follows:

$$\mathcal{S}_{k+1}^{g(i_k^{z*})} = \mathcal{S}_k^{g(i_k^{z*})} \setminus \{i_k^{z*}\}, \quad \forall z \in \mathcal{C}_k, \quad (10a)$$

$$\mathcal{S}_{k+1}^{\ell_k^{z*}} = \mathcal{S}_k^{\ell_k^{z*}} \cup \{i_k^{z*}\}, \quad \forall z \in \mathcal{C}_k. \quad (10b)$$

**Theorem 1** *If the initial partitioning  $\mathcal{P}_0 = \{\mathcal{S}_0^1, \dots, \mathcal{S}_0^m\}$  satisfies constraints (4b), and (4c), then these constraints are satisfied by  $\mathcal{P}_k = \{\mathcal{S}_k^1, \dots, \mathcal{S}_k^m\}$  for all  $k \in \mathbb{Z}_{\geq 0}$  under the partitioning updating in (10). ■*

**Proof:** Assuming that the initial partitioning set  $\mathcal{P}_0$  is established such that  $\bigcap_{\ell \in \mathcal{K}} \mathcal{S}_0^\ell = \emptyset$ , and  $\bigcup_{\ell \in \mathcal{K}} \mathcal{S}_0^\ell = \mathcal{S}$ . Then,  $\mathcal{S}_0^{g(i_0^{z*})} \cap \mathcal{S}_0^{\ell_0^{z*}} = \emptyset$  since  $g(i_0^{z*}) \neq \ell_0^{z*}$  according to (9), and due to the fact that  $g(i_0^{z*}), \ell_0^{z*} \in \mathcal{K}$ , then  $\left\{ \bigcup_{\ell \in \mathcal{K} \setminus \{g(i_0^{z*}), \ell_0^{z*}\}} \mathcal{S}_0^\ell \right\} \cup \left\{ \mathcal{S}_0^{g(i_0^{z*})} \cup \mathcal{S}_0^{\ell_0^{z*}} \right\} = \mathcal{S}$ . From (10), obtaining that

$$\begin{aligned} \mathcal{S}_{k+1}^{g(i_k^{z*})} \cup \mathcal{S}_{k+1}^{\ell_k^{z*}} &= \left\{ \mathcal{S}_k^{g(i_k^{z*})} \setminus \{i_k^{z*}\} \right\} \cup \left\{ \{i_k^{z*}\} \cup \mathcal{S}_k^{\ell_k^{z*}} \right\}, \\ &= \mathcal{S}_k^{g(i_k^{z*})} \cup \mathcal{S}_k^{\ell_k^{z*}}, \end{aligned}$$

for all  $z \in \mathcal{C}_k$ . Finally, if  $\mathcal{S}_k^{g(i_k^{z*})} \cap \mathcal{S}_k^{\ell_k^{z*}} = \emptyset$ , for all  $z \in \mathcal{C}_k$ , then  $\left\{ \mathcal{S}_k^{g(i_k^{z*})} \setminus \mathcal{B} \right\} \cap \left\{ \mathcal{S}_k^{\ell_k^{z*}} \cup \mathcal{B} \right\} = \emptyset$ , for any set  $\mathcal{B}$ . Consequently,  $\mathcal{S}_{k+1}^{g(i_k^{z*})} \cap \mathcal{S}_{k+1}^{\ell_k^{z*}} = \emptyset$ . ■

### 3.3 Illustrative Notation Example

This section is focused on presenting an illustrative example associated to the notation of the partitioning algorithm. To this end, consider a communication network represented by the connected graph in Figure 1(a), which is partitioned into three sub-graphs at time instant  $k$  as presented in Figure 1(b), i.e.,  $\mathcal{S} = \{1, \dots, 20\}$ ,  $\mathcal{E} = \{(1, 2), \dots, (20, 16)\}$ , and  $\mathcal{K} = \{1, 2, 3\}$  corresponding to the partitions of color red, blue and green, respectively. Thus, each partition is given by the three graphs presented in Figure 1(c), i.e.,  $\mathcal{P}_k = \{\mathcal{S}_k^1, \mathcal{S}_k^2, \mathcal{S}_k^3\}$ , where  $\mathcal{S}_k^1 = \{1, \dots, 5, 11, 20\}$ ,  $\mathcal{S}_k^2 = \{6, \dots, 10, 12, 13\}$ , and  $\mathcal{S}_k^3 = \{14, \dots, 19\}$ . Hence, the graphs are given by  $\mathcal{G}_k^1 = (\mathcal{S}_k^1, \mathcal{E}_k^1)$ ,  $\mathcal{G}_k^2 = (\mathcal{S}_k^2, \mathcal{E}_k^2)$ , and  $\mathcal{G}_k^3 = (\mathcal{S}_k^3, \mathcal{E}_k^3)$ .

Regarding the four indicators presented in Section 3.1. Indicators are only considered for the links connecting different partitions, i.e., the links presented in Figure 1(d) given by  $\mathcal{E} = \{(3, 14), (5, 15), (9, 11), (11, 12), (11, 13), (20, 16)\}$ . Moreover, Figures 1(e) and 1(f) present both the internal and external nodes corresponding to the first partition according to (6a) and (6b), respectively, i.e.,  $\tilde{\mathcal{S}}_k^1 = \{1, 2, 4\}$  and  $\tilde{\mathcal{S}}_k^1 = \{3, 5, 11, 20\}$ .

The undirected -not necessarily connected- graph denoted by  $\tilde{\mathcal{G}} = (\tilde{\mathcal{S}}, \tilde{\mathcal{E}})$  is constructed from considering the external nodes as presented in Figures 1(g), where  $\tilde{\mathcal{S}}_k = \{1, 2, 3, 4, 5, 11, 13, 14, 16, 20\}$ . There are as many decisions as components in the graph  $\tilde{\mathcal{G}}$ . By naming arbitrarily the component with nodes  $\{1, 2, 3, 4, 5, 11, 14\}$  as  $1 \in \mathcal{C}_k$ , and the component with nodes  $\{13, 16, 20\}$  as  $2 \in \mathcal{C}_k$ , there are two components in this particular example. According to Figure 1(h), supposing that, for the first component  $i_k^{1*} = 5$ , and according to (7),  $\mathcal{Q}_{5,k} = \{1, 2\}$ . This latter fact means that node  $5 \in \mathcal{S}_k^3$  can change partition in  $\mathcal{K} \setminus g(5)$ , i.e., either  $1$  or  $2 \in \mathcal{K}$ .

Finally, Figure 1(i) illustrates the evolution of the partitioning algorithm with  $\ell_k^{1*} = 1$ , making  $5 \in \mathcal{S}$  become part of partition  $1 \in \mathcal{K}$ .

## 4 Density-Dependent Games

Throughout this section, a novel game-theoretical approach to solve optimization problems, which can deal with constraints and time-varying information-sharing networks, is presented. To this end, consider an undirected connected graph  $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathbf{A})$ , where  $\mathcal{S} = \{1, \dots, n\}$  corresponds to the set of strategies in the population. The set of edges representing the information sharing and/or interaction among agents selecting different strategies is given by  $\mathcal{E} \subset \{(i, j) : i, j \in \mathcal{S}\}$ ; and  $\mathbf{A} = [a_{ij}]$  denotes the  $n \times n$  adjacency matrix where  $a_{ij} = 1$  if  $(i, j) \in \mathcal{E}$ , and  $a_{ij} = 0$  otherwise. Hence, the set of neighbors of a node  $i \in \mathcal{S}$  is denoted by  $\mathcal{N}_i = \{j : (i, j) \in \mathcal{E}\}$ .

Consider a collection of a large and finite number of decision makers forming a population, and which can select a strategy from the set  $\mathcal{S} = \{1, \dots, n\}$ . Thus, let  $p_i \in \mathbb{R}_{\geq 0}$  denote the portion of decision makers selecting the  $i^{\text{th}}$  strategy, and  $\mathbf{p} \in \mathbb{R}_{\geq 0}^n$  be the population state or strategic distribution in the population, where  $\mathbf{p} = [p_1 \dots p_n]^T$ . Different from the classical population games approach [19,23,24], the set of possible population states is now given by the positive orthant, i.e.,  $\Delta = \{\mathbf{p} \in \mathbb{R}_{\geq 0}^n\}$ . Therefore, the population size, which is  $\pi = \sum_{i \in \mathcal{S}} p_i$ , can change along the time since the population is of density-dependent type.

Agents make the decisions in order to maximize their utilities and reproduction chances. These incentives are defined by a fitness function  $f_i(\mathbf{p})$ , for all  $i \in \mathcal{S}$ . The fitness function  $f_i$  takes a strategic distribution and returns an utility corresponding to the  $i^{\text{th}}$  strategy. Thus,  $\mathbf{f}(\mathbf{p})$  denotes the population fitness function  $\mathbf{f} = [f_1 \dots f_n]^T$ , which takes a strategic distribution and returns a vector of utilities. Besides, agents are able to migrate from the  $i^{\text{th}}$  strategy to the  $j^{\text{th}}$  strategy according to a switch rate imposed by a revision protocol  $\varrho_{ij}(\mathbf{f}(\mathbf{p}), \mathbf{p})$ , which determines how decision makers behave [19,25–27].

### 4.1 Distributed Density-dependent Population Games (DDPG)

In order to obtain a density-dependent version of the mean dynamics (Kolmogorov forward equation) [19,25], a reproduction rate denoted by  $\delta_i$ , for all  $i \in \mathcal{S}$  is considered, i.e.,

$$\begin{aligned} \dot{p}_i &= \sum_{j \in \mathcal{N}_i} p_j \varrho_{ji}(\mathbf{f}(\mathbf{p}), \mathbf{p}) - p_i \sum_{j \in \mathcal{N}_i} \varrho_{ij}(\mathbf{f}(\mathbf{p}), \mathbf{p}) \\ &\quad + \delta_i(\mathbf{f}(\mathbf{p}), \mathbf{p}), \quad \forall i \in \mathcal{S}. \end{aligned} \quad (11)$$

Table 1  
Different distributed density-dependent population dynamics

Population dynamics	Differential equation and its compacted form	
D3RD	$\dot{p}_i = p_i \left( f_i(\mathbf{p}) \sum_{j \in \mathcal{N}_i} p_j - \sum_{j \in \mathcal{N}_i} p_j f_j(\mathbf{p}) \right) + \beta_i f_i(\mathbf{p}),$	$\dot{\mathbf{p}} = \mathbf{L}^{(\mathbf{p})} \mathbf{f}(\mathbf{p}) + \text{diag}(\boldsymbol{\beta}) \mathbf{f}(\mathbf{p}).$
D3SD	$\dot{p}_i = \sum_{j \in \mathcal{N}_i} p_j [f_i(\mathbf{p}) - f_j(\mathbf{p})]_+ - p_i \sum_{j \in \mathcal{N}_i} [f_j(\mathbf{p}) - f_i(\mathbf{p})]_+ + \beta_i f_i(\mathbf{p}),$	$\dot{\mathbf{p}} = \tilde{\mathbf{L}}^{(\mathbf{p})} \mathbf{f}(\mathbf{p}) + \text{diag}(\boldsymbol{\beta}) \mathbf{f}(\mathbf{p}).$
D3PD	$\dot{p}_i = \sum_{j \in \mathcal{N}_i} [f_i(\mathbf{p}) - f_j(\mathbf{p})]_+ + \beta_i f_i(\mathbf{p}),$	$\dot{\mathbf{p}} = \mathbf{L} \mathbf{f}(\mathbf{p}) + \text{diag}(\boldsymbol{\beta}) \mathbf{f}(\mathbf{p}).$

Table 2  
Laplacian matrices [25] for the density-dependent population dynamics

Population dynamics	Laplacian forms	
D3RD	$\mathbf{L}^{(\mathbf{p})} = [l_{ij}^{(\mathbf{p})}],$	$l_{ij}^{(\mathbf{p})} = \begin{cases} -a_{ij} p_i p_j, & \text{if } i \neq j, \\ \sum_{r \in \mathcal{S}, r \neq i} a_{ir} p_i p_r, & \text{if } i = j. \end{cases}$
D3SD	$\tilde{\mathbf{L}}^{(\mathbf{p})} = [\tilde{l}_{ij}^{(\mathbf{p})}],$	$\tilde{l}_{ij}^{(\mathbf{p})} = \begin{cases} -\frac{a_{ij}}{2} \left( (1 - \nu_{ij}) p_i + (1 + \nu_{ij}) p_j \right), & \text{if } i \neq j, \\ \sum_{r \in \mathcal{S}, r \neq i} \frac{a_{ir}}{2} \left( (1 - \nu_{ir}) p_i + (1 + \nu_{ir}) p_r \right), & \text{if } i = j, \end{cases}$ where $\nu_{ij} = \text{sgn}(f_i(\mathbf{p}) - f_j(\mathbf{p}))$ .
D3PD	$\mathbf{L} = [l_{ij}],$	$l_{ij} = \begin{cases} -a_{ij}, & \text{if } i \neq j, \\ \sum_{r \in \mathcal{S}, r \neq i} a_{ir}, & \text{if } i = j. \end{cases}$

The interpretation of the reproduction rate is as follows. Let  $\delta_i > 0$ , then it means that there is birth in the  $i^{\text{th}}$  strategy. In the contrary, in case  $\delta_i < 0$ , it implies that there is death (an interpretation of negative reproduction rates) for the  $i^{\text{th}}$  strategy. Therefore, according to this reasoning, notice that  $\delta_i$  must be proportional to the fitness function  $f_i$ , i.e., successful decision makers (those whose fitness functions are greater) should have more chances to have offspring [17].

**Definition 1** A reproduction rate  $\delta : \mathbb{R}^n \times \mathbb{R}_{\geq 0}^n \rightarrow \mathbb{R}^n$  is a function satisfying that, reproduction rates  $\delta_i(\mathbf{f}(\mathbf{p}), \mathbf{p})$  decrease as population  $p_i$  increases [17]. ■

Combining the density-dependent mean dynamics (Kolmogorov forward equation) with reproduction rates  $\delta_i(\mathbf{f}(\mathbf{p}), \mathbf{p}) = \beta_i f_i(\mathbf{p})$ , the migration constraints given by a graph  $\mathcal{G}$ , and different revision protocols  $\varrho_{ij}(\mathbf{f}(\mathbf{p}), \mathbf{p})$ , it is possible to generate multiple density-dependent population dynamics. Using the pairwise proportional imitation protocol  $\varrho_{ij}(\mathbf{f}(\mathbf{p}), \mathbf{p}) = p_j [f_j(\mathbf{p}) - f_i(\mathbf{p})]_+$ , pairwise comparison

protocol  $\varrho_{ij}(\mathbf{f}(\mathbf{p}), \mathbf{p}) = [f_j(\mathbf{p}) - f_i(\mathbf{p})]_+$ , and modified pairwise comparison  $\varrho_{ij}(\mathbf{f}(\mathbf{p}), \mathbf{p}) = [f_j(\mathbf{p}) - f_i(\mathbf{p})]_+ / p_i$  [25], three different density-dependent dynamics are obtained, which are presented in Table 1 together with their respective compacted forms using a Laplacian representation (see Table 2).

Now that some density-dependent population dynamics have been introduced, a stability analysis of the equilibrium point is made. Theorems 2 and 3 show that the mentioned Nash equilibrium  $\mathbf{p}^* \in \Delta$  is asymptotically stable under the density-dependent dynamics in Table 1.

**Theorem 2** Let  $\mathcal{G}$  be connected and the population game  $\mathbf{f}$  be a stable full-potential game, i.e.,  $\nabla V(\mathbf{p}) = \mathbf{f}$ . Then, the Nash equilibrium  $\mathbf{p}^* \in \Delta$  corresponding to a population size  $\pi \in \mathbb{R}_{\geq 0}$  such that  $\mathbf{f}(\mathbf{p}^*) = \mathbf{0}_n$  is asymptotically stable under the D3RD, and the D3SD. ■

**Proof:** Consider the same Lyapunov candidate function as in [25], i.e.,  $E_V(\mathbf{p}) = V(\mathbf{p}^*) - V(\mathbf{p})$ , where  $E_V(\mathbf{p}^*) = 0$ , and  $E_V(\mathbf{p}) > 0$ , for all  $\mathbf{p} \neq \mathbf{p}^*$ . Then, it follows that

$\dot{E}_V(\mathbf{p}) = -(\nabla V(\mathbf{p}))^\top \dot{\mathbf{p}}$ , which is the same as  $\dot{E}_V(\mathbf{p}) = -\mathbf{f}(\mathbf{p})^\top \dot{\mathbf{p}}$ . Now, using  $\dot{\mathbf{p}}$  from the compacted form of the D3RD (see Table 1), it is obtained that

$$\begin{aligned} \dot{E}_V(\mathbf{p}) &= -\mathbf{f}(\mathbf{p})^\top (\mathbf{L}^{(\mathbf{p})} \mathbf{f}(\mathbf{p}) + \text{diag}(\beta) \mathbf{f}(\mathbf{p})), \\ &= -\mathbf{f}(\mathbf{p})^\top \mathbf{L}^{(\mathbf{p})} \mathbf{f}(\mathbf{p}) - \mathbf{f}(\mathbf{p})^\top \text{diag}(\beta) \mathbf{f}(\mathbf{p}). \end{aligned} \quad (12)$$

For the first term in (12), notice that  $\mathbf{L}^{(\mathbf{p})}$  corresponds to the Laplacian of a graph  $\mathcal{G}^{(\mathbf{p})} = (\mathcal{V}, \mathcal{E}, \mathbf{A}^{(\mathbf{p})})$ , where  $\mathbf{A}^{(\mathbf{p})} = [a_{ij}^{(\mathbf{p})}]$  is the adjacency matrix with entries given by  $a_{ij}^{(\mathbf{p})} = a_{ij} p_i p_j$ . The entries of the adjacency matrix are non-negative due to the fact that  $\mathbf{p} \in \Delta$ , which is the positive orthant. Therefore  $\mathbf{L}^{(\mathbf{p})} \geq 0$  for any  $\mathbf{p} \in \Delta$ . Regarding the second term in (12), the diagonal matrix is  $\text{diag}(\beta) \geq 0$  due to the fact that  $\beta_i \geq 0$  for all  $i \in \mathcal{S}$ . Finally, it is concluded that  $\dot{E}_V(\mathbf{p}) \leq 0$ .

Regarding the D3SD, it is obtained that  $\dot{E}_V(\mathbf{p}) = -\mathbf{f}(\mathbf{p})^\top \tilde{\mathbf{L}}^{(\mathbf{p})} \mathbf{f}(\mathbf{p}) - \mathbf{f}(\mathbf{p})^\top \text{diag}(\beta) \mathbf{f}(\mathbf{p})$ . Analysis is the same, but with  $\tilde{\mathbf{L}}^{(\mathbf{p})}$  corresponding to the Laplacian matrix of the graph  $\mathcal{G}^{(\mathbf{p})} = (\mathcal{V}, \mathcal{E}, \mathbf{A}^{(\mathbf{p})})$  with  $a_{ij}^{(\mathbf{p})} = (a_{ij}/2)((1-\nu_{ij})p_i + (1+\nu_{ij})p_j)$ . Therefore  $\tilde{\mathbf{L}}^{(\mathbf{p})} \geq 0$  for any  $\mathbf{p} \in \Delta$ , and it is concluded that  $\dot{E}_V(\mathbf{p}) \leq 0$ . The equality  $\dot{E}_V(\mathbf{p}) = 0$  holds when  $\mathbf{f}(\mathbf{p}) = \mathbf{0}_n$ , and hence  $\mathbf{p}^*$  is asymptotically stable under the D3SD.

Moreover,  $\dot{E}_V(\mathbf{p}) = 0$  in all the cases holds when  $\mathbf{f}(\mathbf{p}) = \mathbf{0}_n$ , and therefore  $\mathbf{p}^*$  is asymptotically stable under the D3RD, and the D3SD with region of attraction in the positive orthant  $\Delta$ . ■

**Theorem 3** *Let  $\mathbf{f}(\mathbf{p}) = \nabla V(\mathbf{p})$ , being  $V(\mathbf{p})$  a strictly concave function, and let  $\mathbf{p}^* \in \mathbb{R}^n$  be an equilibrium point such that  $\mathbf{f}(\mathbf{p}^*) = \mathbf{0}_n$ . If the population-interaction structure is given by a connected graph  $\mathcal{G}$ , then  $\mathbf{p}^* \in \mathbb{R}^n$  is globally asymptotically stable under the D3PD.* ■

**Proof:** Considered the same Lyapunov candidate function as in Theorem 2, i.e.,  $E_V(\mathbf{p}) = V(\mathbf{p}^*) - V(\mathbf{p})$ , where  $E_V(\mathbf{p}^*) = 0$ , and  $E_V(\mathbf{p}) > 0$ , for all  $\mathbf{p} \neq \mathbf{p}^*$ . Then, it follows that  $\dot{E}_V(\mathbf{p}) = -(\nabla V(\mathbf{p}))^\top \dot{\mathbf{p}}$ , which is the same as  $\dot{E}_V(\mathbf{p}) = -\mathbf{f}(\mathbf{p})^\top \dot{\mathbf{p}}$ . Now, replacing  $\dot{\mathbf{p}}$  from the compacted form of the D3PD (see Table 1), it is obtained that

$$\begin{aligned} \dot{E}_V(\mathbf{p}) &= -\mathbf{f}(\mathbf{p})^\top (\mathbf{L} \mathbf{f}(\mathbf{p}) + \text{diag}(\beta) \mathbf{f}(\mathbf{p})), \\ &= -\mathbf{f}(\mathbf{p})^\top \mathbf{L} \mathbf{f}(\mathbf{p}) - \mathbf{f}(\mathbf{p})^\top \text{diag}(\beta) \mathbf{f}(\mathbf{p}). \end{aligned} \quad (13)$$

The first term in (13) is negative since  $\mathbf{L} \geq 0$  for a connected graph  $\mathcal{G}$ . Moreover, the second term in (13) is also negative since the diagonal matrix is  $\text{diag}(\beta) \geq 0$  due to the fact that  $\beta_i \geq 0$  for all  $i \in \mathcal{S}$ . Therefore, it is concluded that  $\dot{E}_V(\mathbf{p}) \leq 0$ . The equality  $\dot{E}_V(\mathbf{p}) = 0$  holds

when  $\mathbf{f}(\mathbf{p}) = \mathbf{0}_n$ , and therefore  $\mathbf{p}^*$  is globally asymptotically stable under the D3PD with initial condition in  $\mathbb{R}^n$  since  $E_V(\mathbf{p})$  is radially unbounded. ■

**Corollary 1** *The asymptotic stability of  $\mathbf{p}^* \in \Delta$  under the D3RD, and the D3SD; and  $\mathbf{p}^* \in \mathbb{R}^n$  under the D3PD stated in Theorem 2 and 3 hold for connected time-varying graphs  $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t), \mathbf{A}(t))$ , i.e., for time-varying neighborhood  $\mathcal{N}_i(t)$ , for all  $i \in \mathcal{S}$ . This statement is concluded since the postulated Lyapunov function  $E_V(\mathbf{p}) = V(\mathbf{p}^*) - V(\mathbf{p})$  is a common function for all possible connected-graph topologies.* ■

#### 4.2 Solving Constrained Optimization Problems with DDPG

Consider a quadratic programming (QP) optimization problem of the form

$$\max_{\mathbf{y}} f(\mathbf{y}), \quad (14a)$$

$$\text{s. t. } \mathbf{E} \mathbf{y} \leq \mathbf{e}, \quad (14b)$$

$$\mathbf{G} \mathbf{y} = \mathbf{g}, \quad (14c)$$

$$\mathbf{y} \in \mathbb{R}_{\geq 0}^v, \quad (14d)$$

where  $f : \mathbb{R}_{\geq 0}^v \rightarrow \mathbb{R}$  is concave, and continuously differentiable. Moreover,  $\mathbf{E} \in \mathbb{R}^{q \times v}$ , and  $\mathbf{e} \in \mathbb{R}^q$  construct the  $q$  inequality constraints (14b), and  $\mathbf{G} \in \mathbb{R}^{r \times v}$ , and  $\mathbf{g} \in \mathbb{R}^r$  construct the  $r$  equality constraints (14c). Inequality constraints can be transformed into equality constraints by adding non-negative slack variables denoted by  $\mathbf{s} \in \mathbb{R}_{\geq 0}^q$ . To this end, consider the vector variable  $\boldsymbol{\xi} = [\mathbf{y}^\top \ \mathbf{s}^\top]^\top \in \mathbb{R}^p$ , where  $p = v + q$ , then the QP optimization problem (14) is reformulated as follows:

$$\max_{\boldsymbol{\xi}} f(\boldsymbol{\xi}), \quad (15a)$$

$$\text{s. t. } \mathbf{H} \boldsymbol{\xi} = \mathbf{h}, \quad (15b)$$

$$\boldsymbol{\xi} \in \mathbb{R}_{\geq 0}^p, \quad (15c)$$

where  $f : \mathbb{R}_{\geq 0}^p \rightarrow \mathbb{R}$  is concave, and continuously differentiable. The matrix  $\mathbf{H} \in \mathbb{R}^{w \times p}$ , and  $\mathbf{h} \in \mathbb{R}^w$  construct the  $w$  equality constraints (15b), where  $w = q + r$ . Now, omitting the positiveness constraints (15c), the Lagrangian function  $L : \mathbb{R}^p \times \mathbb{R}^w \rightarrow \mathbb{R}$  is

$$L(\boldsymbol{\xi}, \boldsymbol{\mu}) = f(\boldsymbol{\xi}) + \boldsymbol{\mu}^\top (\mathbf{H} \boldsymbol{\xi} - \mathbf{h}), \quad (16)$$

where  $\boldsymbol{\mu} \in \mathbb{R}^w$  corresponds to the Lagrange multipliers associated to the  $w$  equality constraints of (15). Moreover,  $\nabla_{\boldsymbol{\xi}} L(\boldsymbol{\xi}, \boldsymbol{\mu}) = \nabla f(\boldsymbol{\xi}) + \mathbf{H}^\top \boldsymbol{\mu}$ , and  $-\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\xi}, \boldsymbol{\mu}) = -\mathbf{H} \boldsymbol{\xi} + \mathbf{h}$ . The Lagrange condition is used to find the possible extreme points  $\boldsymbol{\xi}^* \in \mathbb{R}^p$  of the function  $f(\boldsymbol{\xi})$  (maximum of the function  $f$ ) subject to constraints (15b), in which

$$\begin{bmatrix} \nabla_{\boldsymbol{\xi}} L(\boldsymbol{\xi}^*, \boldsymbol{\mu}^*) \\ -\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\xi}^*, \boldsymbol{\mu}^*) \end{bmatrix} = \mathbf{0}_{p+w}. \quad (17)$$

Now, let  $\mathbf{p} = [\boldsymbol{\xi}^\top \quad \boldsymbol{\mu}^\top]^\top \in \mathbb{R}^n$  be the vector representing the amount of agents in a strategic interaction with  $\mathcal{S} = \{1, \dots, n\}$ , where  $n = p + w$ . Besides, let

$$\mathbf{f}(\mathbf{p}) = [\nabla_{\boldsymbol{\xi}} L(\boldsymbol{\xi}, \boldsymbol{\mu})^\top \quad -\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\xi}, \boldsymbol{\mu})^\top]^\top, \quad (18)$$

be the fitness functions corresponding to all the strategies  $\mathcal{S}$ . The population game (18) can be seen as two different potential games  $\mathbf{f}_{\boldsymbol{\xi}}(\boldsymbol{\xi}, \boldsymbol{\mu}) = \nabla_{\boldsymbol{\xi}} L(\boldsymbol{\xi}, \boldsymbol{\mu})$ , and  $\mathbf{f}_{\boldsymbol{\mu}}(\boldsymbol{\xi}, \boldsymbol{\mu}) = -\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\xi}, \boldsymbol{\mu})$ , whose potential functions are  $L(\boldsymbol{\xi}, \boldsymbol{\mu})$ , and  $-L(\boldsymbol{\xi}, \boldsymbol{\mu})$ , respectively. Therefore, notice that the Hessian of the potential functions is  $\nabla_{\boldsymbol{\xi}}^2 L(\boldsymbol{\xi}, \boldsymbol{\mu}) = \nabla^2 f(\boldsymbol{\xi})$ , and  $\nabla_{\boldsymbol{\mu}}^2 (-L(\boldsymbol{\xi}, \boldsymbol{\mu})) = \mathbf{0}$ . Therefore,  $\mathbf{f}_{\boldsymbol{\xi}}(\boldsymbol{\xi}, \boldsymbol{\mu})$ , and  $\mathbf{f}_{\boldsymbol{\mu}}(\boldsymbol{\xi}, \boldsymbol{\mu})$  are stable games [19]. Finally, since  $\mathbf{f}(\mathbf{p})$  is a full-potential and stable game, and according to Theorem 2 and Corollary 1, the optimization problem (14) can be solved in a distributed way by using the D3RD, the D3SD, or the D3PD, and under time-varying graphs  $\mathcal{G}(t)$ .

**Remark 2** *Population dynamics only allow strategic distributions belonging to the positive orthant  $\Delta$ . However, an optimization problem might require to consider negative values. In this case, it is possible to apply a change of variables in the fitness functions, e.g., in order to consider a constraint of the form  $\underline{\mathbf{y}} \leq \mathbf{y} \leq \bar{\mathbf{y}}$ , being  $\underline{\mathbf{y}} < \mathbf{0}$  and  $\bar{\mathbf{y}} > \mathbf{0}$  a lower and upper bounds, respectively, let apply the change of variables as  $\mathbf{0} \leq \mathbf{y} - \underline{\mathbf{y}} \leq \bar{\mathbf{y}} - \underline{\mathbf{y}}$ , or equivalently,  $\mathbf{0} \leq \tilde{\mathbf{y}} \leq \bar{\mathbf{y}} - \underline{\mathbf{y}}$ . Formulating the problem in the required form shown in (14).* ■

Remark 2 shows that the proposed method with DDPG is versatile to solve optimization problems with different types of inequality constraints in a distributed manner. As an application, next section presents a DMPC controller design based on the D3RD, the D3SD, or the D3PD.

**Remark 3** *There are some considerations to take into account, i.e.,*

- *In order to solve the optimization problem with the Lagrangian function, there are dynamics associated to the Lagrange multipliers. Moreover, it is possible that the Lagrange multipliers get negative values. This fact represents a problem for the D3RD and for the D3SD since those dynamics can only evolve in the positive orthant  $\Delta$  according to Theorem 2. Therefore, to solve the optimization problem using the D3RD and the D3SD, a change of variable must be made to establish an offset. On the other hand, the fact that the Lagrange multipliers can get negative values it is not an inconvenient for the D3PD since trajectories evolve in  $\mathbb{R}^n$  (see Theorem 3).*
- *Regarding the difference between the D3RD and the D3SD, notice that when the reproduction rate is null, then the D3RD preserve the extinction property as in the classical replicator dynamics equation, i.e.,  $\dot{p}_i = 0$ ,*

Table 3

Summary of the three different control approaches (Scenarios).

Control Approach	Communication Graph		Partitioning	
	Constant	Time-varying	Static	Dynamic
Scenario 1	x	✓	x	x
Scenario 2	✓	x	✓	x
Scenario 3	x	✓	x	✓

*for all  $i \in \mathcal{S} \setminus \text{supp}(x)$ , being  $\text{supp}(x) = \{i \in \mathcal{S} : x_i > 0\}$ . In contrast, there is no extinction under the D3SD, not even if the reproductipn rate vanishes.*

*The aforementioned characteristics help to select the most appropriate evolutionary dynamics depending on the structure of the optimization problem.* ■

## 5 Control Approaches

Three alternative control approaches (three different Scenarios) may be designed by using the proposed methodology since the system partitioning algorithm and the consideration of time-varying information-sharing network can be combined in different ways (see the summary in Table 3).

### 5.1 DMPC Controller based on DDPG with Time-varying Information-sharing Network

Consider the general optimization problem behind an MPC controller in (3) such that it can be represented by a QP problem as follows:

$$\min_{\mathbf{U}_k} \mathbf{U}_k^\top \boldsymbol{\Phi} \mathbf{U}_k + \boldsymbol{\phi}_k^\top \mathbf{U}_k, \quad (19)$$

with constraints of the form as in (14), and where  $\mathbf{U}_k = [\mathbf{u}_{k|k}^\top \quad \mathbf{u}_{k+1|k}^\top \quad \dots \quad \mathbf{u}_{k+N-1|k}^\top]^\top$ . Moreover, for the optimization problem (19) the vectors  $\mathbf{e}_k$  and  $\mathbf{g}_k$ , which construct the inequality and equality constraints, vary every iteration  $k$ .

Therefore, notice that the QP formulation (19) for the MPC controller is of the same form as the optimization problem in (14). Furthermore, by adding non-negative slack variables  $\mathbf{s} \in \mathbb{R}_{\geq 0}^q$ , the cost function (19a) is rewritten, and the constraints (19b), and (19c) can be com-

pacted, i.e.,

$$\min_{\xi} \underbrace{\begin{bmatrix} \mathbf{U}_k^\top & \mathbf{s}^\top \end{bmatrix}}_{\xi_k^\top} \underbrace{\begin{bmatrix} \Phi & \mathbf{0}_{n_u \times q} \\ \mathbf{0}_{q \times n_u} & \mathbf{0}_{q \times q} \end{bmatrix}}_{\Psi} \underbrace{\begin{bmatrix} \mathbf{U}_k \\ \mathbf{s} \end{bmatrix}}_{\xi_k} + \underbrace{\begin{bmatrix} \phi_k^\top & \mathbf{0}_q^\top \end{bmatrix}}_{\psi_k^\top} \underbrace{\begin{bmatrix} \mathbf{U}_k \\ \mathbf{s} \end{bmatrix}}_{\xi_k}, \quad (20)$$

$$\text{s. t. } \underbrace{\begin{bmatrix} \mathbf{E} & \mathbb{I}_q \\ \mathbf{G} & \mathbf{0}_{r \times q} \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} \mathbf{U}_k \\ \mathbf{s} \end{bmatrix}}_{\xi_k} = \underbrace{\begin{bmatrix} \mathbf{e}_k \\ \mathbf{g}_k \end{bmatrix}}_{\mathbf{h}_k}. \quad (21)$$

Having added the slack variables, the optimization problem behind the MPC controller is formulated of the form (15), and it can be solved in a distributed manner by using the D3RD, the D3SD, or the D3PD, as explained in Section 4.1.

Regarding the information dependence, it is mainly given by the fitness functions coupling. In order to determine the information-sharing structure for the D3RD, the D3SD, or the D3PD, the fitness functions (18) are expressed in the form

$$\mathbf{f}(\mathbf{p}) = \begin{bmatrix} f_1(\mathbf{p}) \\ \vdots \\ f_n(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1(\mathbf{p}) \\ \mathbf{f}_2(\mathbf{p}) \\ \mathbf{f}_3(\mathbf{p}) \end{bmatrix}. \quad (22)$$

First block  $\mathbf{f}_1(\mathbf{p}) = [\frac{\partial}{\partial p_1} L(\mathbf{p}) \quad \dots \quad \frac{\partial}{\partial p_{n_u}} L(\mathbf{p})]^\top$  of the vector  $\mathbf{f}(\mathbf{p})$  in (22) corresponds to the  $n_u$  variables  $\Delta u$ , second block corresponds to the  $q$  inequality constraints, i.e.,  $\mathbf{f}_2(\mathbf{p}) = -[\frac{\partial}{\partial p_{(n_u+1)}} L(\mathbf{p}) \quad \dots \quad \frac{\partial}{\partial p_{(n_u+q)}} L(\mathbf{p})]^\top$ , and third block corresponds to the  $r$  equality constraints, i.e.,  $\mathbf{f}_3(\mathbf{p}) = -[\frac{\partial}{\partial p_{(n_u+q+1)}} L(\mathbf{p}) \quad \dots \quad \frac{\partial}{\partial p_{(n_u+q+r)}} L(\mathbf{p})]^\top$ .

**Assumption 1** *In the population game  $\mathbf{f}$ , convergence to the Nash equilibrium under the D3RD, the D3SD, or the D3PD, is achieved in shorter time than the sampling time  $\Delta t$  for the discrete system in (1).* ■

In order to check from which strategies it is necessary to get information, the Hessian matrix of the Lagrangian function  $\Theta = [\theta_{ij}] = \nabla^2 L(\mathbf{p})$  is computed, i.e.,  $\theta_{ij} = \nabla_{p_j} f_i(\mathbf{p})$ . Then, the biggest required information-sharing matrix denoted by  $\tilde{\Theta} = [\tilde{\theta}_{ij}]$ , which corresponds to the existing coupling among the portion of agents at each strategy, is given by

$$\tilde{\theta}_{ij} = \begin{cases} 1, & \text{if } \theta_{ij} \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

For the time-invariant graph  $\mathcal{G}$  scenario, the adjacency matrix is given by the biggest required information-sharing matrix, i.e.,  $\mathbf{A} = \tilde{\Theta}$ . Nevertheless, conditions

over the adjacency matrix can be added in order to use less information-sharing links when conveniently. It is highlighted that conditions can be versatile and the time-varying graph can be addressed in different ways besides the one proposed in this paper. In this paper, it is proposed to have an information-sharing graph topology depending on the necessary active constraints. This is made since, under some system state conditions, it is not necessary to consider the whole set of constraints. In this regard, some of them can be properly neglected to reduce both the size of the information-sharing network and the computational burden.

Regions describe *non-safe* sectors in the feasible set, or sectors near limits of a constraint.

- Upper region for states:  $\bar{R}_i^x = \{x_i : \bar{g}_i^x \leq x_i \leq x_i^{\max}\}$ ,
- Lower region for states:  $\underline{R}_i^x = \{x_i : x_i^{\min} \leq x_i \leq \underline{g}_i^x\}$ ,
- Upper region for inputs:  $\bar{R}_j^u = \{u_j : \bar{g}_j^u \leq u_j \leq u_j^{\max}\}$ ,
- Lower region for inputs:  $\underline{R}_j^u = \{u_j : u_j^{\min} \leq u_j \leq \underline{g}_j^u\}$ ,

for all  $i = 1, \dots, n_x$ , and  $j = 1, \dots, n_u$ . Besides, binary variables  $\tilde{\gamma}_{i,k}^x, \underline{\gamma}_{i,k}^x$ , which indicate whether or not the current state  $x_{i,k}$  belongs to a region, are as follows:

$$\tilde{\gamma}_{i,k}^x = \begin{cases} 1, & \text{if } x_{i,k} \in \bar{R}_i^x \\ 0, & \text{otherwise,} \end{cases} \quad \underline{\gamma}_{i,k}^x = \begin{cases} 1, & \text{if } x_{i,k} \in \underline{R}_i^x \\ 0, & \text{otherwise,} \end{cases}$$

where  $i = 1, \dots, n_x$ . Parameters  $\tilde{\gamma}_{j,k}^u$ , and  $\underline{\gamma}_{j,k}^u$ , indicating whether or not the current control input  $u_{i,k}$  belongs to a given region, are stated similarly for  $j = 1, \dots, n_u$ . These binary variables lead to a vector that determines the active and non-active constraints for states and control inputs at each time instant, i.e.,

$$\mathbf{\Gamma}_k^u = [\tilde{\gamma}_{1,k}^u, \dots, \tilde{\gamma}_{n_u,k}^u, \underline{\gamma}_{1,k}^u, \dots, \underline{\gamma}_{n_u,k}^u]^\top, \\ \mathbf{\Gamma}_k^x = [\tilde{\gamma}_{1,k}^x, \dots, \tilde{\gamma}_{n_x,k}^x, \underline{\gamma}_{1,k}^x, \dots, \underline{\gamma}_{n_x,k}^x]^\top.$$

Then, let  $\tilde{\Gamma}_k$  be the diagonal matrix of the active constraints at instant time  $k \in \mathbb{Z}_{\geq 0}$ , i.e.,

$$\tilde{\Gamma}_k = \text{diag}([\mathbf{1}_{n_u}^\top \quad \mathbf{\Gamma}_k^{u^\top} \quad \mathbf{\Gamma}_k^{x^\top} \quad \mathbf{1}_r^\top]).$$

Finally, these conditions over the active constraints lead to a time-varying graph with adjacency matrix  $\mathbf{A}(t)$  that varies its topology every  $\Delta t$ , given by  $\mathbf{A}(k\Delta t) = \tilde{\Gamma}_k \tilde{\Theta} \tilde{\Gamma}_k$ , and the topology corresponding to  $\mathbf{A}(k\Delta t)$  is maintained during a time  $\Delta t$ , i.e.,  $\mathbf{A}(t) = \mathbf{A}(k\Delta t)$ , for all  $k\Delta t \leq t < (k+1)\Delta t$ . Alternatively, the time-varying adjacency matrix can be denoted that

$$\mathbf{A}_k = \tilde{\Gamma}_k \tilde{\Theta} \tilde{\Gamma}_k. \quad (24)$$

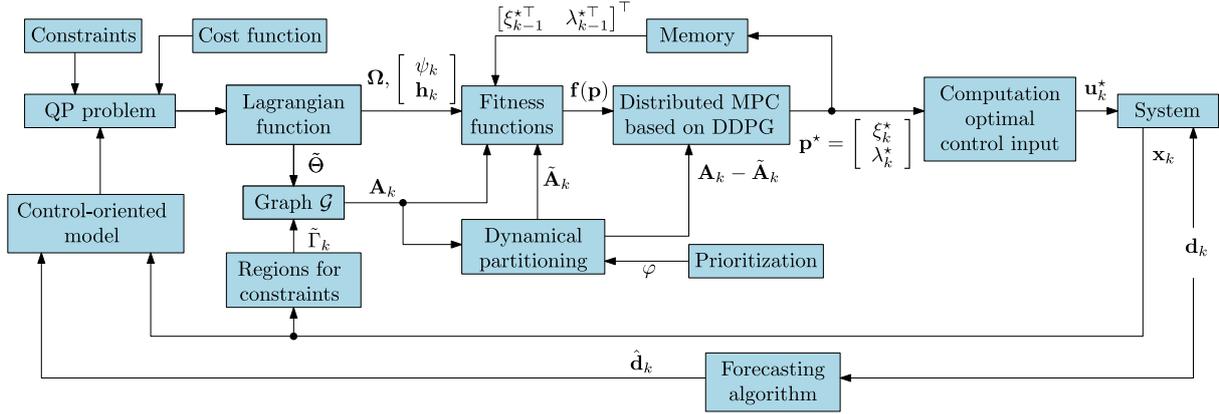


Fig. 2. Summary of the DMPC controller with distributed DDPG and dynamical partitioning.

## 5.2 Partitioning for DMPC Controllers based on DDPG

This section presents how to apply both a static off-line and dynamic on-line system partitioning for the design of a DMPC controller based on DDPG. The first step in order to apply the proposed distributed system partitioning algorithm is to determine the information-sharing graph. This graph is computed as discussed in (23), i.e., the adjacency matrix that determines the graph  $\mathcal{G}$  is given by the required information-sharing matrix, i.e.,  $\mathbf{A} = \tilde{\Theta}$ . Recalling the optimization problem behind the MPC controller in (20), the corresponding Lagrangian function is as follows:

$$L(\boldsymbol{\xi}_k, \boldsymbol{\lambda}_k) = \boldsymbol{\xi}_k^\top \boldsymbol{\Psi} \boldsymbol{\xi}_k + \boldsymbol{\psi}_k^\top \boldsymbol{\xi}_k + \boldsymbol{\lambda}_k^\top (\mathbf{H} \boldsymbol{\xi}_k - \mathbf{h}_k). \quad (25)$$

Therefore, the Karush-Kuhn-Tucker conditions are obtained from  $\nabla_{\boldsymbol{\xi}_k} L(\boldsymbol{\xi}_k^*, \boldsymbol{\lambda}_k^*) = \mathbf{0}$ , and  $\nabla_{\boldsymbol{\lambda}_k} L(\boldsymbol{\xi}_k^*, \boldsymbol{\lambda}_k^*) = \mathbf{0}$ . When the information-sharing graph  $\mathcal{G}$  with adjacency matrix  $\mathbf{A}$  is available, i.e., when there is not partitioning in the system, then the fitness functions are given by the following expression corresponding to (18):

$$\mathbf{f}(\mathbf{p}) = \underbrace{\begin{bmatrix} 2\boldsymbol{\Psi} & \mathbf{H}^\top \\ -\mathbf{H} & \mathbf{0}_{(q+r) \times (q+r)} \end{bmatrix}}_{\boldsymbol{\Omega}} \underbrace{\begin{bmatrix} \boldsymbol{\xi}_k^* \\ \boldsymbol{\lambda}_k^* \end{bmatrix}}_{\mathbf{p}^*} + \begin{bmatrix} \boldsymbol{\psi}_k \\ \mathbf{h}_k \end{bmatrix}. \quad (26)$$

Moreover, when the system is partitioned, there are two classifications for the information-sharing links. Notice that when the partitioning is performed, then there are information-sharing links within the same partition and links connecting different partitions. According to the partitioning approach presented in Section 3.2, an optimal partition  $\mathcal{P}^*$  has an associated graph (not necessarily connected) denoted by  $\tilde{\mathcal{G}}$ . The graph  $\tilde{\mathcal{G}}$  represents the graph whose links  $\tilde{\mathcal{E}}$  corresponds to the links connecting different partitions. The optimization problem behind the DMPC controller is solved by capturing informa-

tion from other partitions throughout the information-sharing links  $\tilde{\mathcal{E}}$ . Then, these links  $\tilde{\mathcal{E}}$  are disconnected, and the DDPG evolve independently at each partition satisfying the information-sharing graph now imposed with an adjacency matrix  $\tilde{\mathbf{A}}$  as presented in (27). Therefore, the amount of required information-sharing links along the time is reduced since the links  $\tilde{\mathcal{E}}$  are not needed for all the time. Hence, the fitness functions are computed as follows:

$$\mathbf{f}(\mathbf{p}) = \underbrace{\left[ (\mathbf{A} - \tilde{\mathbf{A}}) \circ \boldsymbol{\Omega} \right]}_{\tilde{\mathbf{A}}} \underbrace{\begin{bmatrix} \boldsymbol{\xi}_k \\ \boldsymbol{\lambda}_k \end{bmatrix}}_{\mathbf{p}} + \underbrace{\left[ \tilde{\mathbf{A}} \circ \boldsymbol{\Omega} \right]}_{\tilde{\mathbf{A}}} \underbrace{\begin{bmatrix} \boldsymbol{\xi}_{k-1}^* \\ \boldsymbol{\lambda}_{k-1}^* \end{bmatrix}}_{\mathbf{p}^*} + \begin{bmatrix} \boldsymbol{\psi}_k \\ \mathbf{h}_k \end{bmatrix}. \quad (27)$$

Once the equilibrium point  $\mathbf{p}^*$  for the DDPG is obtained, then the procedure can be repeated in order to find a better solution for the optimization problem, i.e., the values corresponding to the information provided throughout the links  $\tilde{\mathcal{E}}$  are updated. In this regard, information from other partitions can be updated to improve solution, and in fact, the same solution that is gotten without partitioning can be obtained repeating update until  $[\boldsymbol{\xi}_k^{*\top} \ \boldsymbol{\lambda}_k^{*\top}]^\top = [\boldsymbol{\xi}_{k-1}^{*\top} \ \boldsymbol{\lambda}_{k-1}^{*\top}]^\top$ . Moreover, in order to avoid infeasibility in the DMPC controller based on DDPG with a system partitioning, more importance is assigned to those links associated to the equality constraints, i.e., constraints in (28e), throughout the relevance matrix  $\mathbf{C}$  as presented in the partitioning algorithm (see Section 3.2). Thus, the optimal system partitioning  $\mathcal{P}^*$  integrates those links within partitions avoiding that they belong to  $\tilde{\mathcal{E}}$ .

Now, consider that the information-sharing graph  $\mathcal{G}$  varies over time as it has been presented in Section 5.1, i.e., that the adjacency matrix  $\mathbf{A}$  varies along the time. Then, due to the fact that the information-sharing graph varies, it is necessary to determine the appropriate partitioning of the system in a dynamical manner such that the partitioning criteria hold. Besides, the set of links  $\tilde{\mathcal{E}}$  also varies along the time, which are links that

are not required for all the time but only to update the fitness functions, i.e., with  $\mathbf{A}_k$ ,  $\tilde{\mathbf{A}}_k$ , and  $\hat{\mathbf{A}}_k$  in (27).

The application of a dynamical system partitioning for a distributed controller has several advantages, i.e., the computational burden is distributed for all the non-centralized controllers for all the sub-systems. When considering that the information-sharing graph varies along the time getting rid of non-active constraints, then the computational burden associated with the optimization problem is reduced. The same system partitioning objectives are still considered although the information-sharing graph varies along the time assigning the appropriate set of not required links  $\tilde{\mathcal{E}}$ . Finally, suppose that there is any inconvenient or fault at any partition. For the conventional centralized MPC controller approach, the whole system is affected. In contrast, when adopting a partitioned non-centralized MPC controller approach, any inconvenient at a partition is decoupled from other. Figure 2 shows the summary of the DMPC with DDPG and with dynamical system partitioning.

## 6 Case Study, Results and Discussion

The Barcelona Water Supply Network (BWSN) is an LSS composed by  $n_x$  tanks,  $n_u$  control inputs (valves and pumps),  $n_s$  drinking water sources, and  $n_d$  water demands as reported in [13]. State vector  $\mathbf{x} \in \mathbb{R}^{n_x}$  is associated to the volumes in tanks, the vector of control inputs  $\mathbf{u} \in \mathbb{R}^{n_u}$  is associated to the manipulated flows throughout valves and pumps, and the vector of disturbances  $\mathbf{d} \in \mathbb{R}^{n_d}$  is associated to the water-demanded flows. The corresponding discrete-time model is the one presented in (1) and its sampling time is  $\Delta t = 1$  hour. Then, Assumption 1 is not strong. The description of the static mass balance at junction nodes in the network is given by (28e). The control objectives are associated to meet a reference assigned by the company in charge of the system, which determines a safety level to satisfy the time-varying demand, and associated to the minimization of the variation of the flows in order to avoid abrupt changes, which might cause damage in the network. These two objectives are considered in the cost function (28a). The BWSN is controlled with an MPC controller whose optimization problem is stated as follows [13]:

$$\begin{aligned} \min_{\mathbf{u}_k, \dots, \mathbf{u}_{k+N-1}} J(\mathbf{x}, \mathbf{u}) &= \sum_{j=1}^N \|\mathbf{x}_{k+j|k} - \mathbf{x}_r\|_{\tilde{\mathbf{Q}}} + \sum_{j=1}^N \|\Delta \mathbf{u}_{k+j|k}\|_{\tilde{\mathbf{R}}} \\ &+ \sum_{j=1}^N \gamma |(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_{2,k+j})^\top \mathbf{u}_{k+j-1|k}|, \end{aligned} \quad (28a)$$

s. t.

$$\mathbf{x}_{k+j+1|k} = \mathbf{A}_d \mathbf{x}_{k+j|k} + \mathbf{B} \mathbf{u}_{k+j|k} + \mathbf{B}_d \mathbf{d}_{k+j|k}, \quad (28b)$$

$$\mathbf{u}_{k+j|k} \in \mathcal{U}, \quad (28c)$$

$$\mathbf{x}_{k+j|k} \in \mathcal{X}, \quad (28d)$$

$$\mathbf{0}_r = \mathbf{E}_u \mathbf{u}_{k+j|k} + \mathbf{E}_d \mathbf{d}_{k+j|k}, \quad (28e)$$

for all  $j \in [0, N-1] \cap \mathbb{Z}_{\geq 0}$  for (28b), (28c), and (28e), and for all  $j \in [1, N] \cap \mathbb{Z}_{\geq 0}$  for (28d); where  $\mathbf{x}_r \in \mathbb{R}^{n_x}$  is a constant desired set-point for the system states  $\mathbf{x} \in \mathbb{R}^{n_x}$ . Moreover,  $\boldsymbol{\alpha}_1 \in \mathbb{R}^{n_u}$  represents the time-invariant costs associated to the water resource, and  $\boldsymbol{\alpha}_2 \in \mathbb{R}^{n_u}$  represents the time-varying costs associated to the operation of valves and pumps. On the other hand,  $\mathbf{E}_u \in \mathbb{R}^{r \times n_u}$ , and  $\mathbf{E}_d \in \mathbb{R}^{r \times n_d}$  construct the  $r$  equality constraints in (28e). The matrices  $\tilde{\mathbf{Q}} \in \mathbb{R}^{n_x \times n_x}$  and  $\tilde{\mathbf{R}} \in \mathbb{R}^{n_u \times n_u}$  are weights assigning a prioritization for the control objectives related to the error and to energy slew rate, respectively. The optimization problem behind the MPC controller in (28) can be conveniently re-formulated with a cost function given by

$$\begin{aligned} J &= (\mathbf{X}_k - \mathbf{X}_r)^\top \mathbf{Q} (\mathbf{X}_k - \mathbf{X}_r) + \Delta \mathbf{U}_k^\top \mathbf{R} \Delta \mathbf{U}_k \\ &+ \mathbf{U}_k^\top \boldsymbol{\alpha}, \end{aligned} \quad (29)$$

where  $\boldsymbol{\alpha} = [(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_{2,k+1})^\top \dots (\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_{2,k+N})^\top]^\top$ , and the weighting matrices are  $\mathbf{Q} = \text{diag}([\tilde{\mathbf{Q}} \dots \tilde{\mathbf{Q}}])$  and  $\mathbf{R} = \text{diag}([\tilde{\mathbf{R}} \dots \tilde{\mathbf{R}}])$ . The reference vector along  $N$  is  $\mathbf{X}_r = [\mathbf{x}_r^\top \mathbf{x}_r^\top \dots \mathbf{x}_r^\top]^\top$ , and vectors  $\mathbf{X}_k$ ,  $\mathbf{U}_k$  and  $\Delta \mathbf{U}_k$  are as follows:  $\mathbf{X}_k = [\mathbf{x}_{k+1|k}^\top \mathbf{x}_{k+2|k}^\top \dots \mathbf{x}_{k+N|k}^\top]^\top$ ,  $\mathbf{U}_k = [\mathbf{u}_{k|k}^\top \mathbf{u}_{k+1|k}^\top \dots \mathbf{u}_{k+N-1|k}^\top]^\top$ , and with slew rate  $\Delta \mathbf{U}_k = [\Delta \mathbf{u}_{k|k}^\top \Delta \mathbf{u}_{k+1|k}^\top \dots \Delta \mathbf{u}_{k+N-1|k}^\top]^\top$ , where  $\Delta \mathbf{u}_{k|k} = \mathbf{u}_{k|k} - \mathbf{u}_{k-1|k}$ . Notice that the problem in (29) can be expressed only in terms of  $\mathbf{U}_k$  by applying a change of variable as in [22] thanks to the fact  $\mathbf{u}_{k-1|k}$  is known at time instant  $k$ . Thus, problem (28) is written in the same form as problem in (19).

Consider the general optimization problem behind an MPC controller in (3). Then, some re-formulations over the cost function (3a) and constraints (3b)-(3e) are made in order to show the density-dependent population-games approach as an alternative tool for DMPC controller design. Therefore, the optimization problem in (28) can be written of the form as in (19) and with constraints of the form as in (14). Moreover, for the optimization problem (19) the vectors  $\mathbf{e}_k$  and  $\mathbf{g}_k$ , which construct the inequality and equality constraints, vary every iteration  $k$ . Notice that the optimization problem in (19) is of the same form of the optimization problem presented in Section 5.1. For this case study, it is proposed the design of three DMPC controllers based on DDPG for three different scenarios corresponding to Table 3.

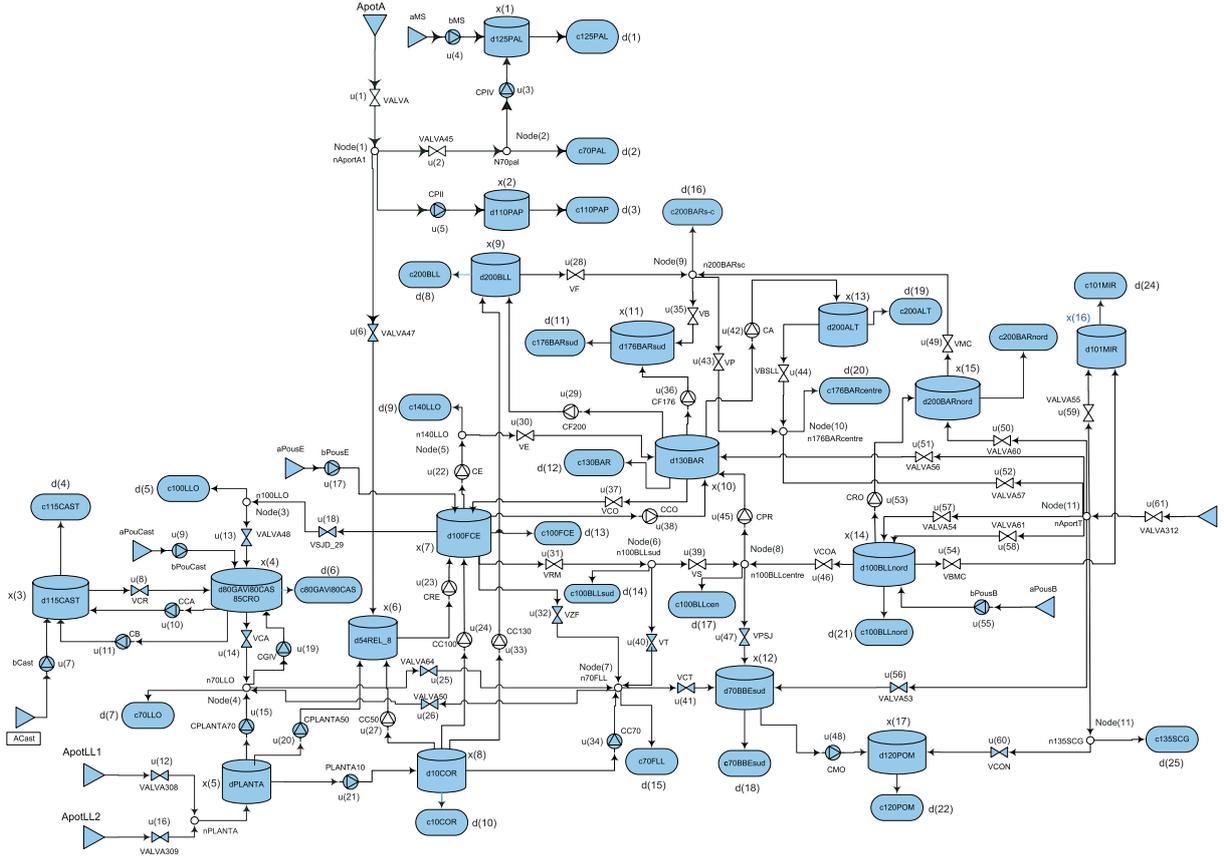


Fig. 3. Case study. BDWN Topology of 17 tanks. Diagram taken from [28].

For all the scenarios and simulation results, the reference has been selected to be  $\mathbf{x}_r = 0.6\mathbf{x}^{\max}$ , and the weights in the cost function are selected to be  $\tilde{\mathbf{Q}} = \mathbf{I}_{n_x}$ ,  $\tilde{\mathbf{R}} = 1000\mathbf{I}_{n_u}$ , and  $\gamma = 1$ .

Figures 5, 6 and 8 present the behavior of the proposed approach for the three scenarios. Figure 5 shows the evolution of some states achieving the imposed reference, also reflecting a proper performance of the proposed distributed density-dependent population-dynamics-based DMPC achieving the references and minimizing abrupt changes in the control signals for all the scenarios. On the other hand, Figure 6 presents the behavior of some control inputs. It can be seen that these control inputs oscillate in order to satisfy the constraints imposed by the demands. That is why these control inputs have the same periodicity as the disturbances (period of 24 hours). Although all the demands have different magnitudes and mean values, they have the same daily periodicity (period of 24 hours) as the disturbances  $d_4, d_8, d_{16}$ , and  $d_{24}$  presented in Figure 7. Figure 8 shows the evolution of the number of connected links in the information-sharing network along the time. It can be seen that, at the beginning, it is needed to have the information-sharing graph corresponding to the biggest required information-sharing matrix for Scenario 1, i.e.,

$\mathbf{A}(t) = \tilde{\Theta}$ . Then, after few iterations, the system reduces the number of required links considerably. This figure also presents the evolution of the links for Scenarios 2 and 3. Moreover, it can be seen a periodic behavior of the number of required links in the information-sharing network, is daily (period of 24 hours) as the disturbances.

In order to compare the performance of the three different scenarios, then two Key Performance Indices (KPIs) are introduced. The first KPI corresponds to the economical costs to operate the actuators, and the second KPI corresponds to the required communication links to perform each control scheme. The two mentioned KPIs are defined as:

- *Economical costs*: these costs correspond to the required energy, and the time-varying water costs during a day, i.e.,

$$\text{KPI}_{\text{costs}}(\text{day}) = \sum_{k=1+24(\text{day}-1)}^{24+24(\text{day}-1)} (\alpha_1 + \alpha_{2,k})^T \mathbf{u}_k. \quad (30)$$

- *Communication costs*: these costs correspond to the required permanent information-sharing links to com-

pute the control inputs during a day, i.e.,

$$\text{KPI}_{\text{costs}}(\text{day}) = \sum_{k=1+24(\text{day}-1)}^{24+24(\text{day}-1)} \frac{(\mathbf{1}_n^\top \mathbf{M}_k \mathbf{1}_n)}{2}, \quad (31)$$

where  $\mathbf{M}_k = \mathbf{A}_k$  for Scenario 1,  $\mathbf{M}_k = \mathbf{A} - \tilde{\mathbf{A}}$  for Scenario 2 (constant), and  $\mathbf{M}_k = \mathbf{A}_k - \tilde{\mathbf{A}}_k$  for Scenario 3.

For the Scenario 1, the regions to determine the active-ness of constraints are computed by using the following parameters:  $\bar{g}_i^x = \underline{g}_i^x = 0.6x_i^{\max}$ , for all  $i = 1, \dots, n_x$ ; and  $\bar{g}_j^u = 0.65u_j^{\max}$ , and  $\underline{g}_j^u = 0.35u_j^{\max}$ , for all  $j = 1, \dots, n_u$ .

Figure 4(a) presents the information-sharing topology required to solve the optimization problem by using the D3RD, the D3SD, or the D3PD, when all the inequality constraints are active, i.e.,  $\mathbf{A}(t) = \Theta$ . Figure 4(b) corresponds to the graph when adopting time-varying graphs and the proposed distributed density-dependent population-dynamics-based DMPC controller. Notice that the graph in Figure 4(d) has a reduction of 36.15% of the information-sharing links. This reduction in the number of links from Figure 4(a) to Figure 4(d) is produced thanks to the fact that at  $k = 77$ , there are non-active inequality constraints. Furthermore, Figure 4(b) and Figure 4(c) correspond to time instants  $k = 33$  and  $k = 11$ , respectively.

For the Scenario 2, it is proposed to use the information-sharing graph  $\mathcal{G}$ , which is shown in Figure 9(a). The partitioning algorithm is performed with weights  $\varphi = [0.5 \ 0.2 \ 0.2 \ 1]^\top$ , and with the parameter  $\kappa = 0$  for the algorithm ending-up condition. The optimal system partitioning  $\mathcal{P}^*$  is the one presented in Figure 9(a). It is important to highlight that the total number of information-sharing links in order to compute the optimal control input according to problem (28) is  $(\mathbf{1}_n^\top \mathbf{A} \mathbf{1}_n)/2 = 361$ . Furthermore, the optimal system partitioning  $\mathcal{P}^*$  has 13 links among partitions, which is the 3.6% of the total number of information-sharing links, representing reduced communication dependence among different partitions, which is desired for the design of non-centralized controllers. With the optimal system partitioning  $\mathcal{P}^*$  presented in Figure 9(a), the information-sharing graph is interpreted or translated into the physical system, obtaining the physical partitioning into  $m$  sub-systems presented in Figure 9(b) (the indices of the  $m$  sub-systems are given by the set  $\mathcal{K}$ ). With the  $m$ -partitioning, a local MPC controller is designed for each sub-system, identifying the information dependence among them as in [13].

Finally, in the Scenario 3, the system partitioning algorithm is performed with arbitrary weights  $\varphi = [1 \ 0.26 \ 0.1 \ 0.5]^\top$ , and with the parameter  $\kappa = 0$  for the algorithm-stop condition. Figures 10 presents the time-varying information-sharing graphs, the corresponding optimal system partitioning highlighting

Table 4  
Summary of KPIs corresponding to operation of actuators and communication links.

Day	Scenario 1		Scenario 2		Scenario 3	
	KPI <sub>costs</sub>	KPI <sub>links</sub>	KPI <sub>costs</sub>	KPI <sub>links</sub>	KPI <sub>costs</sub>	KPI <sub>links</sub>
1	24.2456	10724	22.6773	14736	24.2953	10797
2	21.6782	9800	19.6910	14736	22.2863	9666
3	21.4634	9828	18.2795	14736	22.9248	9671
4	21.3294	9838	17.8871	14736	22.5787	9565
5	21.2401	9840	18.4463	14736	22.9660	9667
<b>Total</b>	109.9567	50030	96.9812	73680	115.0512	49366
<b>Overall</b>	50139.95		73776.98		49481.05	

the links connecting different partitions, and the corresponding physical partitioning for the BWSN. Figure 10(a) corresponds to a unique time instant, i.e.,  $k = 15$ , whereas Figure 10(b) corresponds to 28 time instants, being the most frequent partitioning.

Table 4 shows the KPIs corresponding to the costs associated to each actuator, and to the required communication links. It can be seen that the lowest economical costs  $\text{KPI}_{\text{costs}} = 96.9812$  are obtained with the Scenario 2 corresponding to fixed constraints, i.e., with constant information-sharing network and constant static system partitioning. However, notice that the Scenario 2 also corresponds to the highest costs associated to the communication links  $\text{KPI}_{\text{links}} = 73680$ . In contrast, Scenario 3 is the one with lowest communication costs, i.e.,  $\text{KPI}_{\text{links}} = 49366$ . Moreover, the Scenario 1 has lower economical costs  $\text{KPI}_{\text{costs}} = 109.9567$  in comparison to  $\text{KPI}_{\text{costs}} = 115.0512$  of the Scenario 3. In conclusion, notice that Scenario 3 is the control strategy at the overall as shown in Table 4 if equal relevance is assigned to both KPIs or if more relevance is assigned to the communication links.

## 7 Concluding Remarks

A multi-objective partitioning procedure considering several aspects such as the amount of links connecting different partitions, the size of partitions, the distance among elements, and the importance of links has been presented in order to determine the appropriate partitions in an LSS. As one of the most relevant features of the proposed partitioning is that it can be performed in a distributed manner. Therefore, the DMPC controller based on DDPG is combined with the distributed partitioning algorithm in two different manners, i.e., with static and dynamical system partitioning. The results for these two DMPC controllers based on DDPG and performing both static and dynamical system partitioning are presented, showing the effectiveness of both the DDPG approach and the partitioning for LSS. As further work, the proposed non-centralized control design with partitioning can be tested in presence of faults at some partitions, so that the strategy facilitates their proper isolation.

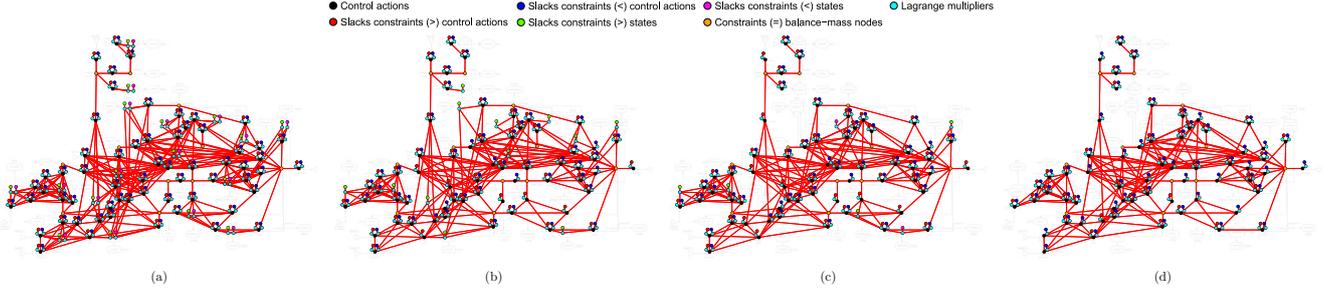


Fig. 4. Communication topologies for the DMPC controller based on DDPG. (a) information-sharing topology for the DMPC controller based on DDPG, i.e.,  $\mathbf{A} = \tilde{\Theta}$ . The number of links in this graph for this case study is 614; and time-varying information-sharing topology for the DMPC controller based on DDPG at time instant: (b)  $k = 33$ , i.e.,  $\mathbf{A}(33\Delta t) = \tilde{\Gamma}(33)\tilde{\Theta}\tilde{\Gamma}(33)$ , (c)  $k = 11$ , i.e.,  $\mathbf{A}(11\Delta t) = \tilde{\Gamma}(11)\tilde{\Theta}\tilde{\Gamma}(11)$ , and (d)  $k = 77$ , i.e.,  $\mathbf{A}(77\Delta t) = \tilde{\Gamma}(77)\tilde{\Theta}\tilde{\Gamma}(77)$ . The number of links in the graphs for this case study are: (b) 468, (c) 435, and (d) 392.

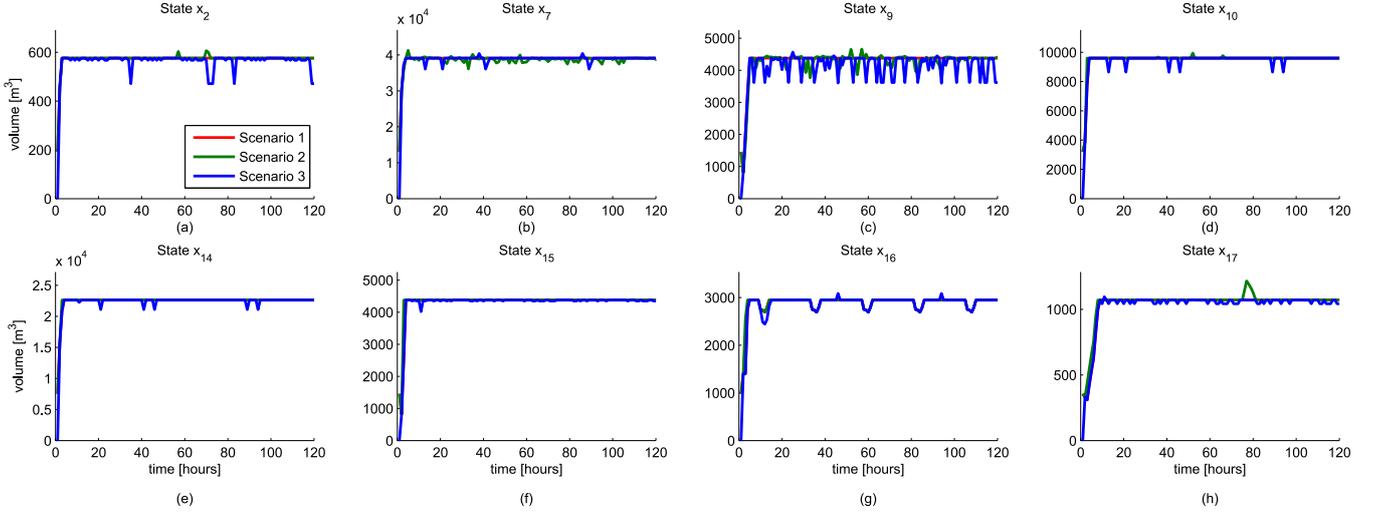


Fig. 5. Evolution of eight system states. Figures (a)-(h) correspond to states  $x_2, x_7, x_9, x_{10}, x_{14}, x_{15}, x_{16}$ , and  $x_{17}$  for the three scenarios.

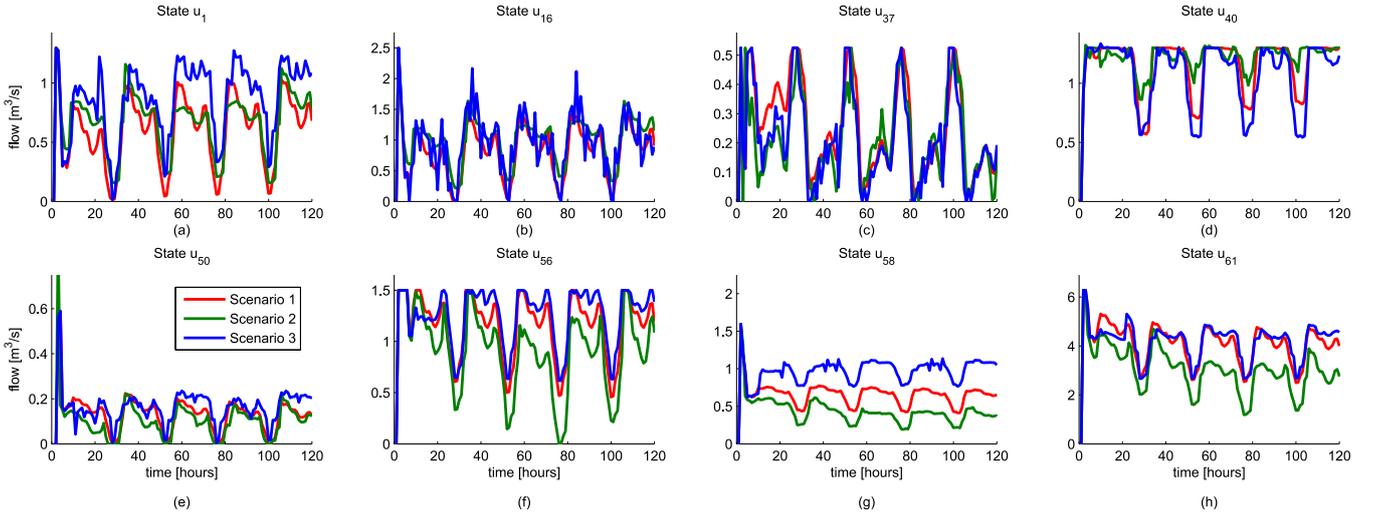


Fig. 6. Evolution of eight control inputs. Figures (a)-(h) correspond to states  $u_1, u_{16}, u_{37}, u_{40}, u_{50}, u_{56}, u_{58}$ , and  $u_{61}$  for the three scenarios.

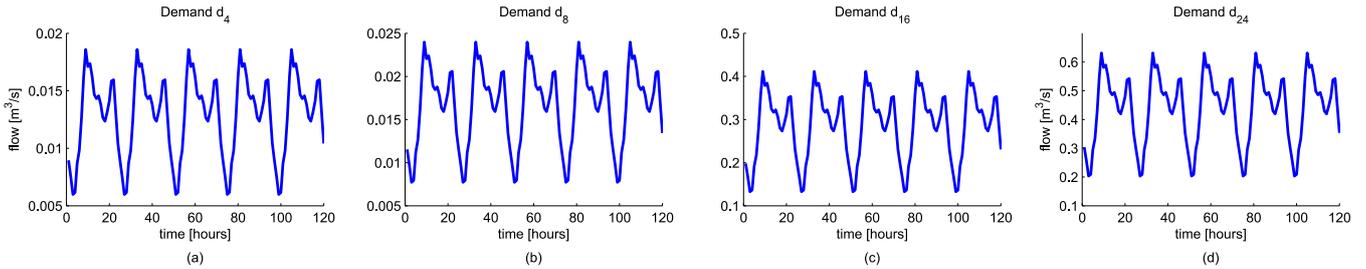


Fig. 7. Evolution of four demands (disturbances). Figures (a)-(d) correspond to states  $d_4$ ,  $d_8$ ,  $d_{16}$ , and  $u_{24}$  for the three scenarios.

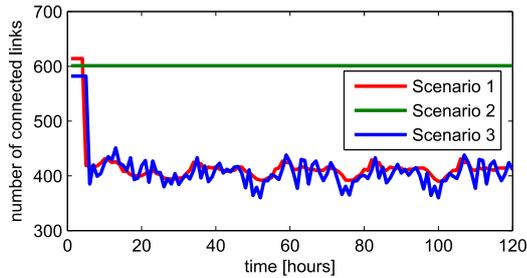


Fig. 8. Evolution of the connected links for the three scenarios.

On the other hand, a general methodology to generate distributed density-dependent population dynamics has been presented by considering a reproduction rate in the distributed mean dynamics. Furthermore, it has been shown the relationship between the equilibrium point of DDPG with the optimal point in a constrained optimization problem by selecting the description of benefits throughout the strategies using the Lagrangian of a potential function. In addition, the asymptotic stability of the equilibrium point under the D3RD, the D3SD, and the D3PD has been formally proven for constant and time-varying population-interaction structures. Then, after introducing this class of dynamics and their properties, they have been applied in the design of a DMPC controller under a time-varying information-sharing network.

## Acknowledgments

Julian Barreiro-Gomez gratefully acknowledges support from U.S. Air Force Office of Scientific Research under grant number FA9550-17-1-0259. The work of Carlos Ocampo-Martinez is partially supported by the project DEOCS (Ref. DPI2016-76493-C3-3-R) from the Spanish MINECO/FEDER. Authors also thank the second stage of the project Pavco-Mexichem-Colciencias.

## References

[1] E. Camponogara, D. Jia, B. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1):44–52, 2002.

[2] P. D. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computer & Chemical Engineering*, 51:21–41, 2013.

[3] R. R. Negenborn and J. M. Maestre. Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, 34(4):87–97, 2014.

[4] R. Scattolini. Architectures for distributed and hierarchical model predictive control – A review. *Journal of process control*, 5:723–731, 19.

[5] D. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(2014):2967–2986, 2014.

[6] A. Bemporad and D. Barcelli. Decentralized model predictive control. In A. Bemporad, M. Heemels, and M. Johansson, editors, *Networked Control Systems, volume 406 of Lecture Notes in Control and Information Sciences*, pages 149–178. Springer London, 2010.

[7] H. Cui and E. W. Jacobsen. Performance limitations on decentralized control. *Journal of Process Control*, 12:485–494, 2002.

[8] J. B. Rawlings and B. T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control*, 18:839–845, 2008.

[9] V. Chandan and A. Alleyne. Optimal partitioning for the decentralized thermal control of buildings. *IEEE Transactions on Control Systems Technology*, 21(5):1756–1770, 2013.

[10] M. R. Kleinberg, K. Miu, N. Segal, H. Lehmann, and T. R. Figura. A partitioning method for distributed capacitor control of electric power distribution systems. *IEEE Transactions on Power Systems*, 29(2):637–644, 2014.

[11] M. Nayeripour, H. Fallahzadeh-Abarghouei, E. Waffenschmidt, and S. Hasanvand. Coordinated online voltage management of distributed generation using network partitioning. *Electric Power Systems Research*, 141(2016):202–209, 2016.

[12] L. Xie, X. Cai, J. Chen, and H. Su. GA based decomposition of large scale distributed model predictive control systems. *Control Engineering Practice*, 57(2016):111–125, 2016.

[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*, 21(2011):775–786, 2011.

[14] J. Marden and J. Shamma. Game theory and distributed control. *Handbook of Game Theory with Economic Applications*, 4:861–899, 2015.

[15] N. Quijano, C. Ocampo-Martinez, J. Barreiro-Gomez, G. Obando, A. Pantoja, and E. Mojica-Nava. The role of population games and evolutionary dynamics in distributed control systems. *IEEE Control Systems*, 37(1):70–97, 2017.

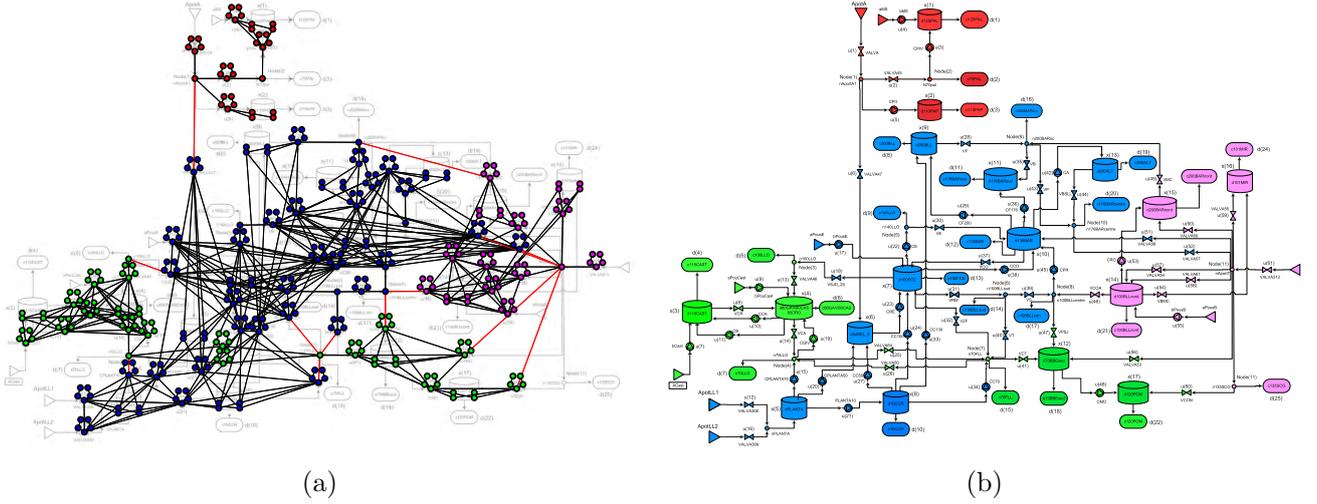


Fig. 9. (a) Information-sharing graph  $\mathcal{G}$  of the BDWN. The optimal partitioning is also presented (partitions 1 – 4 with colors green, blue, magenta, and red, respectively). Links within the same partition with black color and links connecting different partitions with red color, and (b) Corresponding physical partitioning.

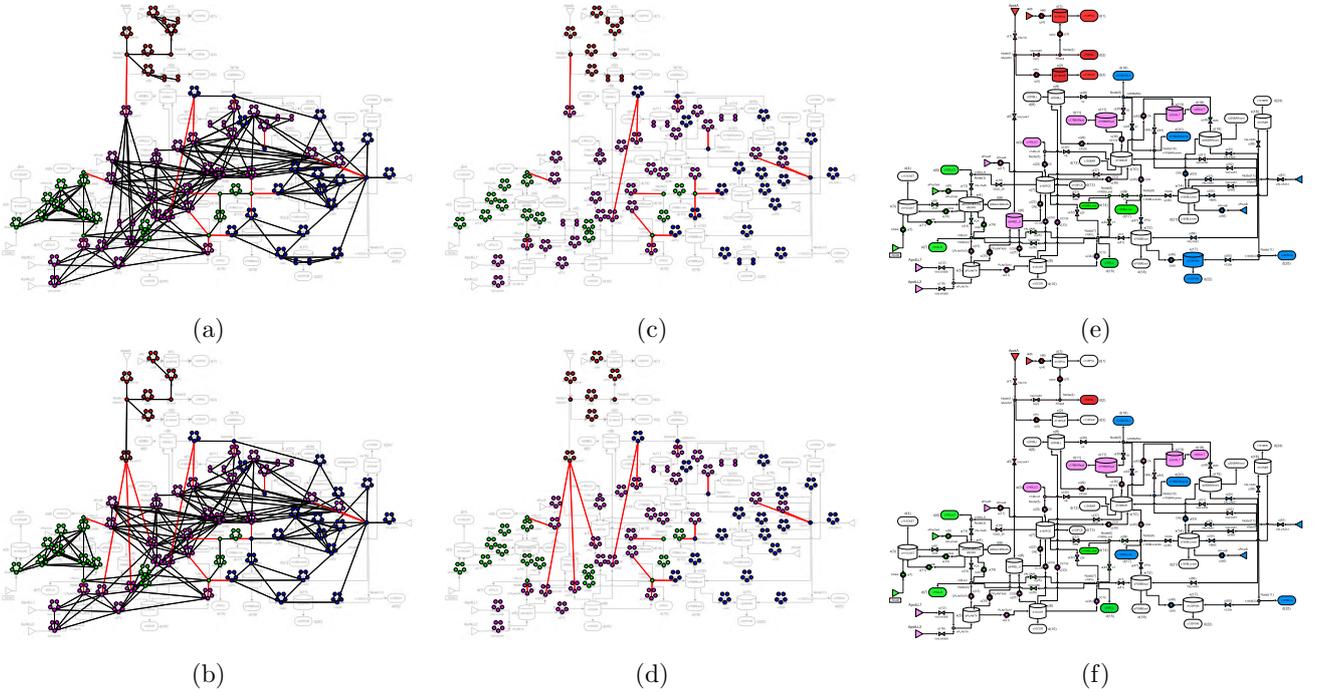


Fig. 10. Partitions for different time instants for the BWSN. Figures (a),(b) correspond to optimal system partitioning  $\mathcal{P}^*$  for the information-sharing graph  $\mathcal{G}$ , Figures (c),(d) correspond to the graph  $\tilde{\mathcal{G}}$  for the partitions, and Figures (e),(f) correspond to the respective physical partitioning. Moreover, Figures (a),(c),(e) correspond to time instant  $k = 15$ , and Figures (b),(d),(f) correspond to time instants  $k \in \{40, \dots, 42, 44, \dots, 47, 64, \dots, 66, 68, \dots, 71, 88, \dots, 90, 92, \dots, 95, 112, \dots, 114, 116, \dots, 119\}$ .

[16] R. Cressman and V. Křivan. Migration dynamics for the ideal free distribution. *The American Naturalist*, 168(3):384–397, 2006.

[17] S. Novak, K. Chatterjee, and M. A. Nowak. Density games. *Journal of Theoretical Biology*, 334(2013):26–34, 2013.

[18] A. Gupta. Fast and effective algorithms for graph partitioning and sparse-matrix ordering. *IBM Journal of Research and Development*, 41(1):171–183, 1997.

[19] W. Sandholm. *Population games and evolutionary dynamics*.

Cambridge, Mass. MIT Press, 2010.

[20] Y. Wang, C. Ocampo-Martinez, and V. Puig. Stochastic model predictive control based on Gaussian processes applied to drinking water networks. *IET Control Theory and Applications*, 10:947–955, 2016.

[21] Y. Wang, C. Ocampo-Martinez, V. Puig, and J. Quevedo. Gaussian-process-based demand forecasting for predictive control of drinking water networks. In *Proceedings of the 9th International Conference on Critical Information*

- Infrastructures Security*, pages 13–15, Limassol, Cyprus, 2014.
- [22] J. Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.
- [23] J. W. Weibull. *Evolutionary Game Theory*. The MIT Press, London, England, 1997.
- [24] H. Tembine, E. Altman, R. El-Azouzi, and Y. Hayel. Evolutionary games in wireless networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(3):634–646, 2010.
- [25] J. Barreiro-Gomez, G. Obando, and N. Quijano. Distributed population dynamics: Optimization and control applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(2):304–314, 2017.
- [26] H. Tembine, E. Altman, R. ElAzouzi, and W. H. Sandholm. Evolutionary game dynamics with migration for hybrid power control in wireless communications. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC)*, pages 4479–4484, Cancun, Mexico, 2008.
- [27] J. Barreiro-Gomez and H. Tembine. Constrained evolutionary games by using a mixture of imitation dynamics. *Automatica*, 97(2018):254–262, 2018.
- [28] J. Barreiro-Gomez, C. Ocampo-Martinez, and N. Quijano. Dynamical tuning for MPC using population games: A water supply network application. *ISA Transactions*, 69(2017):175–186, 2017.