Annual Reviews in Control 61 (2026) 101046

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

Partitioning techniques for non-centralized predictive control: A systematic review and novel theoretical insights☆

Alessandro Riccardi *, Luca Laurenti, Bart De Schutter

Delft Center for Systems and Control (DCSC), Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Keywords:
Partitioning
Model predictive control
Multi-agent systems
Decentralized MPC
Distributed MPC
Hierarchical MPC
Coalitional control
Graph representations
Topology
Network
Hybrid systems
Large-scale systems
Clustering
Community detection
k-means
Problem decomposition
Mixed-integer programming

ABSTRACT

The partitioning problem is of central relevance for designing and implementing non-centralized Model Predictive Control (MPC) strategies for large-scale systems. These control approaches include decentralized MPC, distributed MPC, hierarchical MPC, and coalitional MPC. Partitioning a system for the application of non-centralized MPC consists of finding the best definition of the subsystems, and their allocation into groups for the definition of local controllers, to maximize the relevant performance indicators. The present survey proposes a novel systematization of the partitioning approaches in the literature in five main classes: optimization-based, algorithmic, community-detection-based, game-theoretic-oriented, and heuristic approaches. A unified graph-theoretical formalism, a mathematical re-formulation of the problem in terms of mixed-integer programming, the novel concepts of predictive partitioning and multi-topological representations, and a methodological formulation of quality metrics are developed to support the classification and further developments of the field. We analyze the different classes of partitioning techniques, and we present an overview of their strengths and limitations, which include a technical discussion about the different approaches. Representative case studies are discussed to illustrate the application of partitioning techniques for non-centralized MPC in various sectors, including power systems, water networks, wind farms, chemical processes, transportation systems, communication networks, industrial automation, smart buildings, and cyber–physical systems. An outlook of future challenges completes the survey.

Contents

☆ This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement No. 101018826)—Project CLariNet.

* Corresponding author.
E-mail address: a.riccardi@tudelft.nl (A. Riccardi).

https://doi.org/10.1016/j.arcontrol.2026.101046
Received 14 September 2025; Received in revised form 15 December 2025; Accepted 13 January 2026
Available online 19 January 2026

1. Introduction

1.1. Motivation

Modern systems are increasingly characterized by architectural scales and implementation complexities that challenge the implementation of centralized control strategies (Kordestani et al., 2021; Šiljak, 2008). This trend is supported by the advancements and availability of information transmission networks, as well as by the wide accessibility of computing resources (Kamel et al., 2016). When the scale of a system grows, it is common and advisable to structure it as a collection of autonomous interconnected components (subsystems). These subsystems should coordinate or be coordinated to achieve a common goal. To this aim, these entities necessitate local computing power, and communication and negotiation abilities: this is why, when these features are available, these advanced subsystems are usually defined as control agents. A schematic representation of a network of control agents is proposed in Fig. 1. Consequently, modern systems constituted by multiple agents having scales that exceed specific (hardware) operational thresholds are commonly referred to as large-scale multi-agent systems (LS-MASs) (Dorri et al., 2018). Examples of LS-MASs can be found in infrastructural systems such as power generation and distribution networks (Javid et al., 2024; Kundur & Malik, 2022; Poullikkas, 2013; Rakhshani et al., 2019); urban and freeway networks (Siri et al., 2021); railway and subway networks (Louf et al., 2014); water distribution networks (Bello et al., 2019); oil and gas distribution networks; large groups of mobile robots such as swarms of UAVs (Zhou et al., 2020), or of terrestrial and maritime autonomous vehicles; large plants for chemical processing (Metzger & Polakow, 2011), which might also integrate autonomous energy generation; large industrial networks (Galloway & Hancke, 2013); and satellite constellations (Curzi et al., 2020); where this list of applications keeps growing and evolving with the introduction of new technologies.

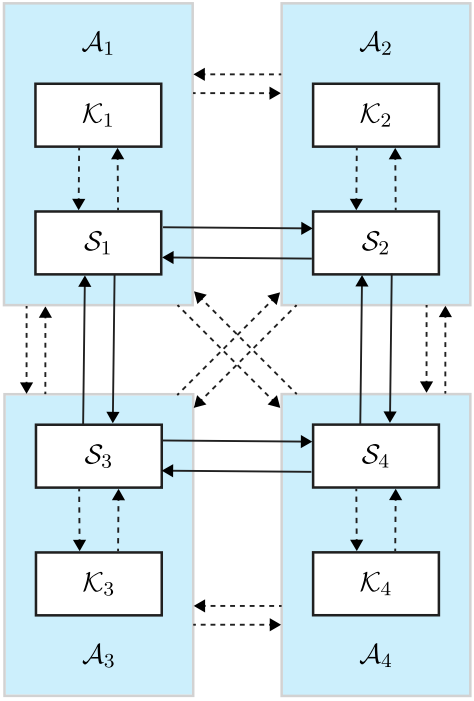


Fig. 1. A network of control agents. Subsystems are indicated by $\mathcal{S}$, local controllers by $\mathcal{K}$, and control agents by $\mathcal{A}$. The solid lines represent the interactions at the physical level, i.e. the dynamical couplings; instead, the dashed lines represent interactions at the information level.

Conventional control methodologies such as proportional–integral control and pole placement (Ogata, 2022), loop-shaping and h-infinity synthesis, Skogestad and Postlethwaite (2001), or feedback linearization (Khalil, 2002) are not directly applicable to LS-MASs because of the presence of a large number of input–output channels and the large spatial distribution of such networks, which complicate centralized controller design and parameter tuning. Therefore, deployment of non-centralized control strategies (Bakule, 2008; Šiljak, 1991) is necessary for LS-MASs, and the level of sophistication of such approaches is tightly related to the availability of reliable communication channels and local computing power.

1.2. Non-centralized MPC: Control architectures

One of the most advanced modern control strategies is model predictive control (MPC) (Rawlings et al., 2017), which integrates the use of a mathematical model of the system dynamics with optimal control methodologies to compute predictive control actions that optimize performance while guaranteeing the stability of the controlled system, as well as the respect of operational constraints (Machowski, 2002; Mesbah, 2016), according to the receding horizon paradigm. The MPC framework has also significantly evolved thanks to its design flexibility, which allows a relatively easy development of non-centralized predictive control strategies (NCen-MPC) (Christofides et al., 2013; Maestre & Negenborn, 2014), i.e. of MPC strategies in which the computation of the control action for the overall system is not performed by a single central unit, but divided across control agents. The traditional classification of these strategies (Scattolini, 2009) comprehends decentralized MPC (Dec-MPC), distributed MPC (DMPC), and hierarchical MPC (HMPC). A conceptual representation of these architectures is proposed in Fig. 2. More recently, a novel NCen-MPC methodology incorporating concepts from game theory has emerged, called coalitional predictive control (Coal-MPC) (Fele et al., 2017). In this survey, we will abbreviate centralized MPC as CMPC, to distinguish it from NCen-MPC. A list of these abbreviations is reported in Table 1. The single common characteristic of all NCen-MPC approaches is that they assume to operate in a network of agents, where, for each individual subsystem, a local optimization problem is solved. Then, the various techniques are distinguished according to how they handle communication and coordination of the local control actions.

When referring to NCen-MPC techniques, the simplest coordination technique is Dec-MPC, in which there is no communication among agents, but the effect of neighboring subsystems on local dynamics is generally assumed to be contained in invariant sets, thus allowing stable operation of such networks while preserving privacy, security, and resilience since there is no information sharing. A communication and coordination protocol is instead at the basis of DMPC approaches, where the agents in the network usually share their measurements or predicted evolution of local variables with neighbors, thus allowing for iterative or non-iterative adjustments of local control actions. In the context of linear systems, this distributed control approach can achieve global performance close to CMPC while drastically reducing computation times, and allowing real-time operation of the networks where centralized predictive control would not be possible. In HMPC, the control architecture is structured across multiple vertical layers, with at least the presence of a global coordinator and a set of local controllers. These strategies pose as an alternative to DMPC, and can enhance global coordination, as well as network resilience, introduce privacy features, or allow for multi-time-scale operation of different network models at different aggregation layers. Finally, the Coal-MPC strategy arises as the result of the combination of predictive control with game theory. In fact, in Coal-MPC, the network is seen as a collection of agents that participate in a cooperative game with the objective of maximizing the global collective outcome, which is the global operation cost of the network.

In conclusion, NCen-MPC strategies allow for the introduction of complex control features, such as advanced algorithmic coordination procedures, plug-and-play capabilities, and privacy and security preservation strategies, into LS-MASs. At the same time, NCen-MPC strategies can ensure stable real-time control of LS-MASs while preserving the optimality of their operation as much as possible.

1.3. The partitioning problem

The underlying assumption of the above discussion about NCen-MPC of LS-MASs is that the network is provided as a collection of agents with full autonomy. While this assumption may seem simple to satisfy, this is not always true in practice. In fact, the definition of the agents themselves may be challenging, especially for large and interconnected networks. Additionally, even if the network is given as a collection of individual agents, it might be more convenient for network operation to aggregate them into bigger entities. These two distinct classes of problems, i.e. the definition of the agents of the network and the problem of their aggregation, fall both into the category of network partitioning (Chanfreut et al., 2021b; Šiljak, 1991).

Formally speaking, the partitioning problem consists of finding the optimal allocation of a group of elements into given sets according to a given metric. If the network $\mathcal{N}$ is provided as a collection of agents $N_{\mathcal{A}}$, i.e. $\mathcal{N} = \{\mathcal{A}_1, \dots, \mathcal{A}_{N_{\mathcal{A}}}\}$, and we have a number N_C of possible sets for the allocation, whose quality is defined by a cost function $h(\cdot)$, then the optimal partitioning problem consists in finding the set $\mathcal{P}$ (i.e. the partition) defined as $\mathcal{P} = \{C_1, \dots, C_{N_C}\}$, where the elements C_i are groups of agents $\mathcal{A}_j$, such that the quality measure $h(\mathcal{P})$ is optimized. On the other hand, if the network $\mathcal{N}$ is provided as a monolithic system that does not show any natural decomposition, the partitioning problem consists of selecting several subsystems $N_{\mathcal{A}}$ for which control agents can be defined, which allows to interpret the network as a collection of agents $\mathcal{N} = \{\mathcal{A}_1, \dots, \mathcal{A}_{N_{\mathcal{A}}}\}$. Also in this case, the subsystem selection is

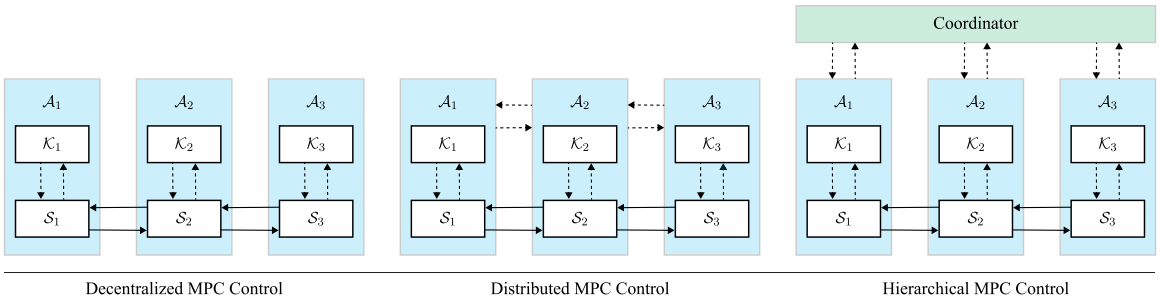


Fig. 2. Main categories of non-centralized control architectures. In decentralized control, there is no information-level interaction among control agents. In distributed control, the information-level interaction is horizontal, i.e. each control agent can communicate with the others. In hierarchical control, the information-level interaction among control agents is vertical, i.e. they should, in principle, communicate with the coordinator. Mixed approaches are also possible.

generally guided by a cost function $h(\cdot)$. Both these problems are known to be NP-hard (Brandes et al., 2006; Karp, 1972; Sandholm et al., 1999).

When the partitioning problem is applied to NCen-MPC, several further features can be developed and extended for both the partitioning and the MPC. Many questions may arise, such as: What is the best definition for the individual agents? How can agents be allocated optimally into sets to maximize the performance of the NCen-MPC architecture? How can the partitioning strategy handle topological changes in the network or different operating conditions? These are a few examples of profound technical challenges that researchers in this field have encountered in the last decades, finding answers and new open problems.

Many of the partitioning strategies that will be presented in this survey are borrowed from other scientific sectors, such as network and graph theory, machine learning, or computer science in general. A general overview of clustering methodologies applied to distributed network control can be found in Chanfreut et al. (2021b), which can serve as a general reference for these methods, while the current survey is tailored specifically for NCen-MPC. We also refer to the work (Xu & Wunsch, 2005) to explore further general clustering methodologies such as k-means, fuzzy c-means, and hierarchical clustering. Other general approaches that have been applied to partitioning for NCen-MPC are community detection methodologies (Fortunato, 2010; Fortunato & Hric, 2016), such as modularity maximization and spectral algorithms; and coalition formation approaches (Apt & Witzel, 2009), which have led to the development of game-theory-based MPC architectures.

1.4. Survey objectives and contributions

Under these considerations, the present survey has two main overarching goals:

1. Unifying in a common framework all the results currently present in the literature addressing the partitioning problem for NCen-MPC.
2. Laying foundations for further systematic developments of this field.

These two objectives are achieved through the following series of steps: a systematization of fundamental notions for graph representation of dynamical systems and networks; the introduction of precise key performance indicators that are comparable across strategies and application domains, as well as a precise assessment methodology of the quality of a partition; a categorization of the known partitioning strategies for NCen-MPC in terms of methodology, partitioning objective, and relative control strategy; a discussion of the main partitioning methodologies to highlight their strengths and limitations; a brief technical discussion of each partitioning technique found in the literature; and a classification of the current application domains of the partitioning techniques.

Table 1
List of abbreviations.

MPC	Model Predictive Control
CMPC	Centralized Model Predictive Control
NCen-MPC	Non-Centralized Model Predictive Control
Dec-MPC	Decentralized Model Predictive Control
DMPC	Distributed Model Predictive Control
HMPC	Hierarchical Model Predictive Control
Coal-MPC	Coalitional Model Predictive Control
NLin-MPC	Nonlinear Model Predictive Control
LS-MAS	Large-Scale Multi-Agent System
MIMO	Multiple-Input Multiple-Output

Further, we extend the analysis and classification of the partitioning techniques with novel theoretical insights, which are: the introduction of multi-topological graph representations to model variable topologies, and their link to hybrid systems; a formal definition of the partitioning problem for performance optimization in terms of a bi-level mixed-integer program (MIP); and a re-definition of the problem of time-varying partitioning, introducing the concept of predictive partitioning for control.

Given the extension of this survey, and the amount of different topics explored in detail, we provide an overview of its organization in Section 2 below, briefly describing the contents and the objectives of each section.

2. Organization of the survey

In this section, we present the structure of the survey, briefly describing the content of each section. This will provide the reader with an organic view of the material presented, and will help to navigate the content, having a general knowledge of all the topics that will be discussed throughout the survey.

Graph representations. Most partitioning techniques, both involving NCen-MPC or other control strategies, are based on abstract representations of the underlying system dynamics (Šiljak, 1991). This representation is generally provided in the form of a graph (Diestel, 2017); therefore, it is natural to start the discussion about partitioning techniques by introducing graph representations in Section 3. In this section, we classify the graph representations used in partitioning and presented in Fig. 3. This classification is supported by a technical discussion of each type of representation in dedicated subsections.

Partitioning for predictive control. Once the abstract representation of the network is available, the partitioning problem for NCen-MPC can be formally introduced and discussed in Section 4. In this section, we discuss the general problem definition and its common characteristics usually present in the partitioning techniques. In addition, we provide metrics and an evaluation methodology to assess the quality of a partition, and we complete the discussion by introducing the novel concept of predictive partitioning as a component of the MPC formulation.

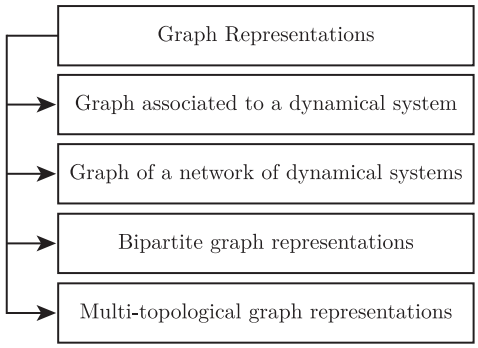


Fig. 3. Graph representations used in partitioning for NCen-MPC.

Classification of the partitioning techniques. In Section 5, we will provide a classification of the partitioning methodologies for the application of NCen-MPC according to three criteria: (1) the general partitioning class; (2) the subclass defined by the main structure of the method or by its objective; and (3) the control architecture to which it has been applied. The classification performed according to the first two criteria is proposed in Fig. 4, where the first level of the classification tree defines the main class, and the second level defines the subclass. The main theoretical characteristics as well as the strengths and limitations of the five main partitioning classes are discussed in Section 5.1, for the subclasses in Section 5.2, and for the methodologies in Section 5.3. Finally, in Section 5.4 we classify the techniques according to the control methodology for which they have been designed.

Analysis of the individual partitioning techniques. Once the classification of the partitioning strategies has been presented, and the main characteristics of each class and subclass have been highlighted, we deepen the technical discussion by providing further details about the methods in each class in Fig. 4. Therefore, an extensive analysis of the individual methodologies in the literature can be found in the dedicated sections, which are: Section 6 for optimization-based partitioning; Section 7 for algorithmic partitioning; Section 8 for community-detection-based partitioning; Section 9 for game-theory-based partitioning; and Section 10 for heuristic partitioning.

Applications. In Section 11, we discuss the main case studies that have been used in the literature about partitioning for NCen-MPC. These are divided by application sector, and, when possible, we also provide reference systems with further details about the systems considered. In addition, we discuss for each application domain how different partitioning methodologies have been used in the literature.

Conclusions and future work. The overall discussion of the main topic of the survey is completed in Section 12 with final considerations about the state of this research field, and with recommendations for future work, identifying the current research gaps and potential new directions to explore.

3. Graph representations

At the basis of almost all partitioning approaches, there is a graph representation of the system to be decomposed. Accordingly, specific graph representations can be deployed when defining a partitioning strategy for applying an NCen-MPC method. These representations belong to three main categories: (1) graphs equivalent to dynamical systems; (2) graphs representing networks of dynamical systems; and (3) graph representations of an optimization problem. In this section, we first introduce graph theory terminology that will be used throughout the article. Then, we present the classes of graphs introduced above. We close the section by conceptually reformulating the graph representation of a network of dynamical systems linking multi-topological graphs and hybrid systems.

3.1. Fundamentals of graph theory

A graph (Diestel, 2017) is an ordered pair of sets $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \dots, n\}$ is the set of n vertices (or nodes), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of the edges (or arcs, links). The edges are associated to the vertices through an $n \times n$ binary adjacency matrix A^{adj}, where $A^{\text{adj}}_{(i,j)} = 1$ if and only if an edge $\epsilon_{ij} = (i, j) \in \mathcal{E}$ exists. Therefore, the topology of the graph is specified by the adjacency matrix A^{adj}, and the set of the edges can also be written as $\mathcal{E} = \{(i, j) \mid [i, j \in \mathcal{V}] \wedge [A^{\text{adj}}_{(i,j)} = 1]\}$. A subgraph of $\mathcal{G}$ is a graph $\mathcal{S}_\ell = (\mathcal{V}_\ell, \mathcal{E}_\ell)$ representing a part of $\mathcal{G}$. The set of vertices $\mathcal{V}_\ell$ is a subset of $\mathcal{V}$, i.e. $\mathcal{V}_\ell \subseteq \mathcal{V}$, and the set of the edges is $\mathcal{E}_\ell = \{(i, j) \mid [i, j \in \mathcal{V}_\ell] \wedge [A^{\text{adj}}_{(i,j)} = 1]\}$, where the topology is still specified by the relevant entries of A^{adj}. For a directed graph $\mathcal{G}$, an edge $\epsilon_{ij} = (i, j)$ denotes an arrow starting from node i and ending in node j. A graph $\mathcal{G}$ is weighted if a weighting matrix W^{adj} assigning to each edge a number is specified in addition to A^{adj}. For each vertex $i \in \mathcal{V}$ we denote by d_i its degree, i.e. the number of edges entering or exiting that vertex. In directed graphs, we can specify an in-degree (d_i^{in}) and an out-degree (d_i^{out}), if the edge is respectively ending or starting in the vertex i. For a vertex i, the neighborhood of i is the set of all vertices connected to it, and we denoted it by $\mathcal{N}_i = \{j \in \mathcal{V} \mid [(i, j) \vee (j, i)] \in \mathcal{E}\}$. For a subgraph $\mathcal{S}_\ell = (\mathcal{V}_\ell, \mathcal{E}_\ell)$, the frontier is its set of nodes that are connected to nodes outside the subgraph, i.e. $\mathcal{F}_\ell = \{i \in \mathcal{V}_\ell \mid [(i, j) \vee (j, i)] \in \mathcal{E}, j \in \mathcal{V} \setminus \mathcal{V}_\ell\}$. These fundamental concepts will be extended throughout the survey for specific topics when necessary.

3.2. Graph associated to a dynamical system

The most direct and intuitive graph representation of a dynamical system is the so-called associated graph. According to Šiljak (1991), the earliest formulation of this type of graph representation for linear systems can be traced back to Lin (1974). We start by presenting associated graph representations for linear discrete-time systems, where the same formulation proposed in Šiljak (1991) for the continuous-time version holds. Consider the dynamics:

$$\mathcal{S} : \begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases}, \tag{1}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$ are respectively the state, input, and output of the system; and A, B, C are matrices of appropriate dimensions. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ associated to (1) is constructed by first defining one node for each variable, which provides the set of vertices $\mathcal{V} = \{x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}, y_1, \dots, y_{n_y}\}$, where this set can be considered as the union of the sets for the individual state, input, and output variables, i.e. $\mathcal{V} = \mathcal{V}_x \cup \mathcal{V}_u \cup \mathcal{V}_y$, $|\mathcal{V}| = n_x + n_u + n_y$. Then, the set of edges $\mathcal{E}$ is built looking at the nonzero entries of matrices A, B, C, and as before, it can be thought of as the union of three different sets $\mathcal{E} = \mathcal{E}_{ux} \cup \mathcal{E}_{xx} \cup \mathcal{E}_{xy}$. These sets of edges define the interactions among variables, and are derived respectively as $\mathcal{E}_{ux} = \{(i, j) \mid i \in \mathcal{V}_u, j \in \mathcal{V}_x, B_{(i,j)} \neq 0\}$, $\mathcal{E}_{xx} = \{(i, j) \mid i, j \in \mathcal{V}_x, A_{(i,j)} \neq 0\}$, $\mathcal{E}_{xy} = \{(i, j) \mid i \in \mathcal{V}_x, j \in \mathcal{V}_y, C_{(i,j)} \neq 0\}$. This graph $\mathcal{G}$ associated with the dynamics (1) is static because the dynamical system is time-invariant. Moreover, the graph represents the interactions among the variables in the system. A measure of this interaction is provided by the weighting matrix that can be constructed considering the entries of matrices A, B, C:

$$W^{\text{adj}} = \begin{bmatrix} A & B & 0 \\ 0 & 0 & 0 \\ C & 0 & 0 \end{bmatrix}. \tag{2}$$

A more recent evolution in the associated graph representation is found in Riccardi et al. (2025c), where the following nonlinear dynamics is considered:

$$\mathcal{S} : \begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = h(x(k)) \end{cases}. \tag{3}$$

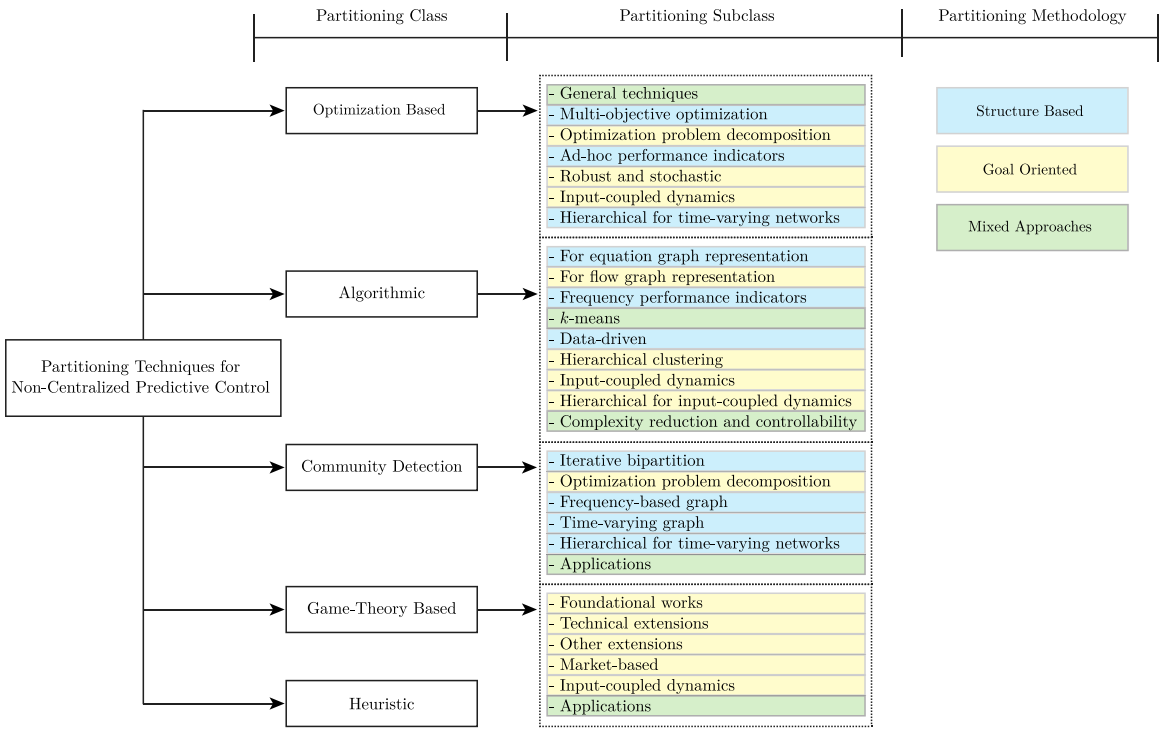


Fig. 4. Categorization of the partitioning techniques in classes and subclasses. The methodologies in each subclass can be further distinguished between the approaches based on the structure of the network, and the ones oriented at achieving a given objective, whether it is a control or another functional specification.

The scope in Riccardi et al. (2025c) is to obtain a weighted and time-varying representation $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$ of the system (3). To this aim, using the same vertices definition introduced for (1), the following weighting function is defined:

$$w_{(i,j)}(k) = \begin{cases} \frac{\partial f_j(x(k),u(k))}{\partial i} & \text{for} \quad i \in \mathcal{V}_u \cup \mathcal{V}_x, j \in \mathcal{V}_x \\ 0 & \text{for} \quad i \in \mathcal{V}, j \in \mathcal{V}_u \\ \frac{\partial h_j(x(k))}{\partial i} & \text{for} \quad i \in \mathcal{V}_x, j \in \mathcal{V}_y \end{cases}. \tag{4}$$

Accordingly, a time-varying set of edges $\mathcal{E}(k)$ is defined as:

$$\mathcal{E}(k) = \{(i,j) \mid i,j \in \mathcal{V}, w_{(i,j)}(k) \neq 0\}. \tag{5}$$

This time-varying graph can capture the instantaneous interactions among the system variables at each time step. In the most general case, a different topological representation exists at each time step. Accordingly, a different choice of graph partition might be the best option for non-centralized predictive control. However, such an approach is computationally demanding.

Example 1. We consider the following linear discrete-time system to show how to construct the graph associated with a dynamical system. Consider the system:

$$x(k+1) = Ax(k) + Bu(k), \tag{6}$$

with $x \in \mathcal{X} \subseteq \mathbb{R}^{10}$, $u \in \mathcal{U} \subseteq \mathbb{R}^3$, where the matrices A and B are defined by the entries

$$\begin{array}{llll} a_{2,1} = 0.5 & a_{6,1} = 0.1 & a_{8,2} = 0.84 & a_{9,2} = 0.57 \\ a_{8,4} = 0.54 & a_{9,5} = 0.91 & a_{2,6} = 0.98 & a_{3,6} = 0.96 \\ a_{5,6} = 0.8 & a_{6,7} = 0.6 & a_{2,8} = 0.31 & b_{4,1} = 0.04 \\ b_{9,1} = 0.6 & b_{10,1} = 0.63 & b_{2,2} = 0.02 & b_{4,2} = 0.6 \\ b_{10,2} = 0.11 & b_{1,3} = 0.19 & b_{2,3} = 0.03, & \end{array} \tag{7}$$

and zero elsewhere. According to the definition of a graph $\mathcal{G}$ associated with a dynamical system, we define the set of vertices $\mathcal{V} = \{u_1, \dots, u_3, x_1, \dots, x_{10}\}$, while the nonzero entries of matrices A, B define the edges in $\mathcal{E}$ of the graph and their weights in the matrix W^{adj}. The representation of this graph is given in Fig. 5. This example will be continued in Section 4.1 to show how to select subsystems for constructing control agents.

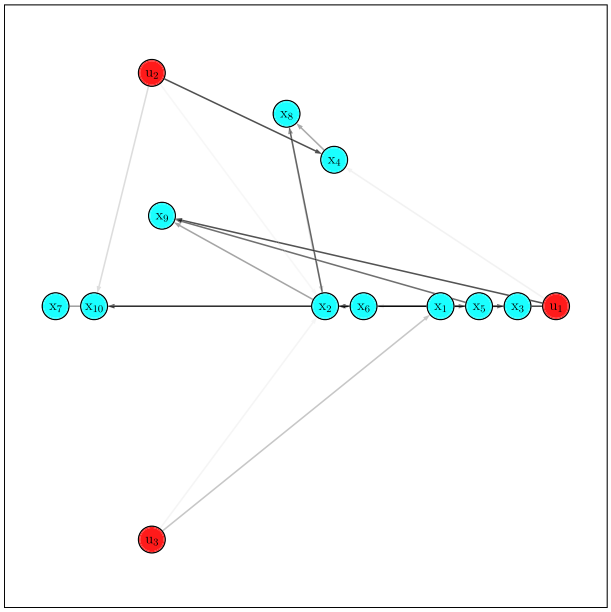


Fig. 5. Graph associated with the dynamical system (7). The vertices are the system variables and are colored in red if they are inputs and cyan if they are states. The arrows represent the edges, and their opacity the strength of interaction, i.e. the weight, defined by the entries of matrices A and B.

Table 2
Randomly generated topology of the network in Fig. 7. The entries $w_{i,j}$ are the ith row and jth column of the weighted adjacency matrix W^{adj}.

$w_{1,25} = 0.53$	$w_{2,3} = 0.36$	$w_{2,12} = 0.01$	$w_{3,33} = 0.60$	$w_{4,26} = 0.41$	$w_{5,31} = 0.47$	$w_{6,38} = 0.32$	$w_{7,33} = 0.24$	$w_{8,19} = 0.24$	$w_{9,49} = 0.20$
$w_{10,40} = 0.36$	$w_{11,35} = 0.72$	$w_{12,2} = 0.01$	$w_{12,10} = 0.42$	$w_{13,7} = 0.17$	$w_{14,44} = 0.44$	$w_{15,31} = 0.67$	$w_{16,7} = 0.46$	$w_{17,28} = 0.42$	$w_{18,40} = 0.76$
$w_{19,14} = 0.67$	$w_{20,31} = 0.55$	$w_{21,34} = 0.37$	$w_{22,4} = 0.66$	$w_{23,1} = 0.20$	$w_{24,47} = 0.51$	$w_{25,46} = 0.78$	$w_{26,41} = 0.10$	$w_{27,40} = 0.60$	$w_{28,22} = 0.35$
$w_{29,47} = 0.43$	$w_{30,46} = 0.16$	$w_{31,13} = 0.68$	$w_{32,15} = 0.34$	$w_{33,10} = 0.66$	$w_{34,29} = 0.19$	$w_{35,6} = 0.43$	$w_{36,33} = 0.60$	$w_{37,7} = 0.41$	$w_{38,36} = 0.40$
$w_{39,46} = 0.23$	$w_{40,36} = 0.44$	$w_{41,35} = 0.31$	$w_{42,39} = 0.66$	$w_{43,38} = 0.39$	$w_{44,29} = 0.19$	$w_{45,39} = 0.49$	$w_{46,21} = 0.69$	$w_{47,16} = 0.40$	$w_{48,12} = 0.29$
$w_{49,12} = 0.13$	$w_{50,40} = 0.77$								

3.3. Graph representation of a network of systems

A different type of graph representation is considered when the dynamical system is a network admitting a natural decomposition into fundamental subsystems interacting through their dynamics. In this case, the network admits a graph representation $\mathcal{G}$ where the individual subsystems constitute the elements of the set of vertices $\mathcal{V} = \{\mathcal{S}_1, \dots, \mathcal{S}_{N_\mathcal{S}}\}$. The set of edges $\mathcal{E}$ is defined by state-to-state interactions. Accordingly, to each subsystem $\mathcal{S}_i$ are associated a local state $x_{\mathcal{S}_i} \in \mathbb{R}^{n_{x_{\mathcal{S}_i}}}$ and input $u_{\mathcal{S}_i} \in \mathbb{R}^{n_{u_{\mathcal{S}_i}}}$. The neighbors of a node of the network, i.e. of a subsystem $\mathcal{S}_i$, is the set $\mathcal{N}_{\mathcal{S}_i} = \{\mathcal{S}_j \mid (i,j) \in \mathcal{E}\}$. The definition of an output vector $y_{\mathcal{S}_i} \in \mathbb{R}^{n_{y_{\mathcal{S}_i}}}$ can also be included, but it will be omitted in the following for simplicity. In other words, for a general nonlinear system of the form (3), there exists a natural subdivision of the state and input vectors such that every individual subsystem is described by:

$$\mathcal{S}_i : x_{\mathcal{S}_i}(k+1) = f_{\mathcal{S}_i}(x_{\mathcal{S}_i}(k), (x_{\mathcal{S}_j}(k))_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}}, u_{\mathcal{S}_i}(k)). \tag{8}$$

This type of representation has been extensively used in partitioning for non-centralized predictive control, especially in the form of linear interacting systems, where each subsystem takes the form:

$$\mathcal{S}_i : \begin{cases} x_{\mathcal{S}_i}(k+1) = A_{\mathcal{S}_i} x_{\mathcal{S}_i}(k) + B_{\mathcal{S}_i} u_{\mathcal{S}_i}(k) + w_{\mathcal{S}_i}(k) \\ w_{\mathcal{S}_i}(k) = \sum_{\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}} A_{\mathcal{S}_{ij}} x_{\mathcal{S}_j}(k) \end{cases}. \tag{9}$$

Each subsystem $\mathcal{S}_i$ is affected only by its local input, and is coupled to its neighbors through dynamic interactions defined by matrices $A_{\mathcal{S}_{ij}}$. This coupling is seen by subsystem $\mathcal{S}_i$ as an exogenous signal $w_{\mathcal{S}_i}$ whose nature is determined by the coordination protocol used in the control strategy, i.e. it is considered a disturbance in decentralized control, or it is known or measurable for cooperative strategies. Further details about this topic are given in Section 3.5 where multi-topological representations are introduced.

Remark 1. From the discussion above, it is clear that each subsystem defined by (8) can itself be seen as a graph as described in Section 3.2. A possible algorithmic approach to link the graph associated with a dynamical system and the graph associated with a network of dynamical systems is proposed in Riccardi et al. (2025c).

Remark 2. In the definition of subsystem (8), we assumed that each $\mathcal{S}_i$ is driven only by its local input $u_{\mathcal{S}_i}$. There is, however, the mathematical possibility that dynamics (8) may be driven also by $u_{\mathcal{S}_j}$ with $\mathcal{S}_j \in \mathcal{N}_{\mathcal{S}_i}$. The resulting networks are constituted by input-coupled subsystems. We decided to treat these networks in separate subsections.

Example 2. In this example, we propose two different network representations of control agents, one having a modular topology, the other having a random one. According to the discussion above, a control agent will incorporate subsystem dynamics and all the control, communication, coordination, and algorithmic requirements for deploying an NCen-MPC strategy.

A network can be considered modular if it exhibits a high level of modularity, which can be quantified using the modularity metric, but also visually because it will present recurring patterns. An example of such a network with 64 control agents is reported in Fig. 6, where the recurring structure of 4 and 16 agents is evident. The topology of this network is defined as follows: from the thickest to the thinnest lines, the bidirectional interactions have a strength of $w_{i,j} = 0.1, 0.01, 0.001$.

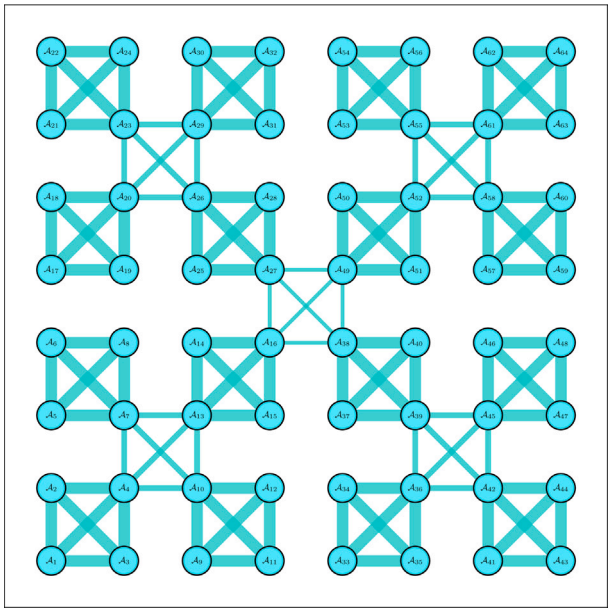


Fig. 6. Graph representation of a modular network with 64 agents. The width of the edges represents the strength of the interaction among the agents. This network exhibits a repeating modular pattern.

The second network has 50 control agents and a randomly generated topology, which is reported in Table 2, and which shows the presence of directed arcs. The network representation is proposed in Fig. 7.

We will use these modular and random networks in Section 4.4 to show an application of optimization-based and algorithmic partitioning approaches and the evaluation methodology for the quality of a partition.

3.4. Bipartite graph representations

In a bipartite graph (Diestel, 2017), the set of the nodes is divided into two groups $\mathcal{V} = \mathcal{V}_a \cup \mathcal{V}_b$, $\mathcal{V}_a \cap \mathcal{V}_b = \emptyset$, and all the edges start in one group and end in another. This type of graph representation has two main use cases in partitioning for non-centralized control. In the first case, a bipartite graph is used to represent the relations between the variables and the constraints of an optimization problem, e.g. as done in Tang, Allman et al. (2018). This approach is used to decompose the optimization problem by minimizing the number of complicating[1] constraints that are removed in the distributed solution of the problem. In

[1] Complicating constraints are those that introduce an interdependence into subproblems, thus affecting (complicating) the separability of the original problem. In this discussion, complicating constraints are those that involve variables of different subsystems.

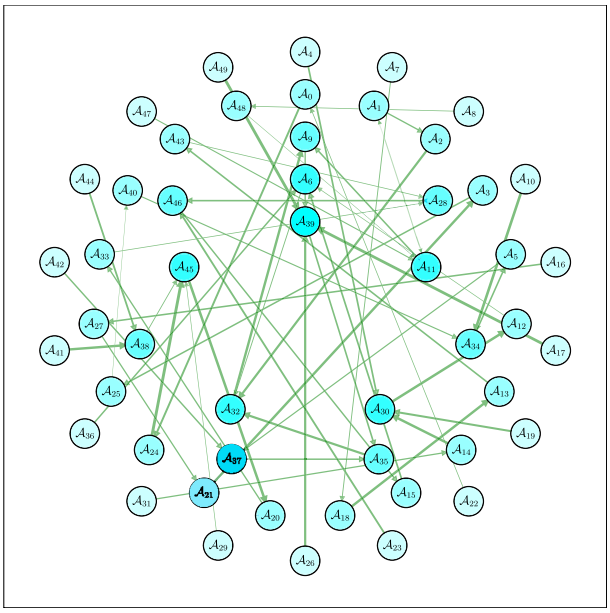


Fig. 7. Graph representation of a random network with 50 agents. The nodes are sorted according to their degree, which is also reflected in the strength of their color. The randomly generated topology is detailed in Table 2.

the second case, a bipartite graph is used to represent the input–output paths of the network, as done in Tang and Daoutidis (2018) and Wang et al. (2023). In these approaches, the relationships between output and input variables are made explicit. Then, among all possible paths between each pair, the shortest is chosen. Accordingly, partitioning is used to minimize the interactions between input and output dynamics, an approach that is conceptually similar to the quantification of input–output interactions in MIMO systems using an RGA matrix (Skogestad & Postlethwaite, 2001).

Example 3. In this example, we propose the use of a bipartite graph representation for a network subject to complicating constraints.[2] Consider the following optimization problem representing MPC optimization at a generic time step k for a network of linear systems, each with two states and one input, with no dynamical coupling, but subject to complicating constraints:

$$\begin{array}{ll} \min\limits_{\tilde{x},\tilde{u}} & J(\tilde{x},\tilde{u}) \\ \text{s.t.} & x(k+1) = Ax(k) + Bu(k) \\ c_1(k): & x_1(k) + x_2(k) \le 0 \\ c_2(k): & x_1(k) + x_3(k) \le 0 \\ c_3(k): & u_1(k) + u_2(k) + x_4(k) + 5 \le 0 \\ c_4(k): & u_2(k) + x_1(k) + x_3(k) \le 0 \end{array}, \tag{10}$$

where $\tilde{x}, \tilde{u}$ represent state and input sequences over an optimization horizon N, and $k = 0, \dots, N$. The constraints c_i introduce interactions among the subsystems of the network, which can be, e.g. interpreted as a set of specifications on shared resources. This coupling can be captured by the bipartite graph in Fig. 8, where the nodes in one set are the variables, the nodes in the other set are the constraints, and the arcs represent the participation of variables in constraints. To partition a network subject to complicating constraints, it is possible to develop algorithmic procedures to maximize the effect of constraint couplings among cooperating agents in the same coalition, and to minimize the coupling between agents in different coalitions. These inter-coalition

[2] The reader can refer to Table 5 for examples of bipartite representations used to capture input–output interactions in MIMO systems.

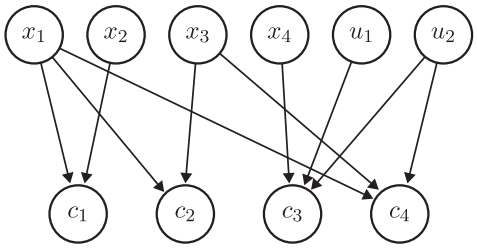


Fig. 8. The bipartite graph representation of a set of complicating constraints. The graph is constituted by two sets of node, one for the optimization variables, one for the problem constraints. The arcs represents the participation of a variable in a constraint. The graph is bipartite because it is formed by two sets of nodes for which arcs only go from one set to another.

constraints can be ignored in solving local problems at first, and they can then be accounted for in a later step of the network optimization.

3.5. Multi-topological network representations

Consider a network of dynamical systems where the connections are determined by a variable topology whose nature will be specified later in this section. The presence of a link introduces a directed relationship between two subsystems that represents a dynamic coupling as described in Fig. 9. Concerning the representation of a network of systems in Section 3.3, here we consider only macro links connecting subsystems, thus omitting the subscript notation related to the topological representation of the interactions among variables of different subsystems. Moreover, we index each subsystem $\mathcal{S}_i$ with the letter i, so that the network of subsystems is made by the set of nodes $\mathcal{V} = \{1, \dots, N_S\}$. These choices simplify the presentation of the following concepts. The existence of a link between the subsystem i and j at a time step k can be represented by the binary variable $\epsilon_{ij}(k)$ such that:

$$\epsilon_{ij}(k) = \begin{cases} 1 & \text{if } \mathcal{S}_i \text{ is connected to } \mathcal{S}_j \text{ at time step } k \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

The collection of these links determines the topology of the network. In the context of control systems, a link representing a dynamical coupling in this network can have three different natures:

- The existence of the link depends on the input-state configuration of the network, i.e. the network has an input-state-dependent topology. This happens when the dynamical coupling is determined by the regions of the input-state configuration of the system, such as in PWA dynamics (Bemporad & Morari, 1999; Heemels et al., 2001).
- The link can be activated or deactivated as a part of the control strategy of the network, i.e. it is a decision variable.
- The link activation is driven by an external function, either known or unknown.

A possible topology can co-exist for each of the above-mentioned link classes. Consequently, the overall topology of the network will result from the composition of these superposed topologies, i.e. a multi-topological network representation, as in Fig. 9.

In the general case, we assume a number of N_ϵ distinct topological levels characterizing the network. We associate a binary variable $\epsilon^q_{ij}(k)$ representing the connection between areas i and j in the topological level q at time step k. According to the nature of the topology with which this variable is associated, it can be an input-state-dependent variable, a decision variable, or a signal. Since all binary variables must be equal to one for a connection to exist, the state of variable $\epsilon_{ij}(k)$ is directly determined by the product:

$$\epsilon_{ij}(k) = \prod_{q=1}^{N_\epsilon} \epsilon^q_{ij}(k). \tag{12}$$

Incorporating binary variables $\epsilon_{ij}(k)$ in the network description is straightforward. For this, consider the network of nonlinear systems:

$$x(k+1) = f(x(k), u(k)), \tag{13}$$

and assume it admits a decomposition in N_S subsystems according to the discussion in Section 3.3. Then, their time-varying topological dynamics is:

$$x_i(k+1) = f_i(x_i(k), u_i(k), \omega_i(k)) \tag{14}$$

$$\omega_{ij}(k) = \epsilon_{ji}(k) x_j(k) \qquad \forall j \in \mathcal{N}_i, \tag{15}$$

where $x_i \in \mathbb{R}^{n_{x_i}}$, $u_i \in \mathbb{R}^{n_{u_i}}$ are the state and input vectors of subsystem i; and the vector ω_i constituted by the elements ω_{ij} incorporates all topologically defined dynamical couplings of subsystem i with the its neighborhood $\mathcal{N}_i$.

3.6. Multi-topological representations and hybrid systems

When applying the concept of multi-topological time-varying representations to networks of linear systems, the result is a hybrid network system (Tabuada, 2009). For the sake of simplicity, and without any loss of generality, in what follows, we consider the case of three topological levels of different nature, but the more general case of $N_\epsilon > 3$ topological levels follows similarly. In particular, the network is described as:

$$x_i(k+1) = A_{ii} x_i(k) + B_{ii} u_i(k) + \omega_i(k) \tag{16}$$

$$\omega_i(k) = \sum_{j \in \mathcal{N}_i} \epsilon_{ij}(k) A_{ij} x_j(k) \tag{17}$$

$$\epsilon_{ij}(k) = \epsilon^1_{ij}(k) \epsilon^2_{ij}(k) \epsilon^3_{ij}(k) \tag{18}$$

$$\text{s.t.} \quad \epsilon^1_{ij}(k) = 1 \Leftrightarrow \begin{bmatrix} x_j(k) \\ u_j(k) \end{bmatrix} \in \Omega^{\mathrm{a}}_j, \tag{19}$$

where $A_{ii} \in \mathbb{R}^{n_{x_i} \times n_{x_i}}$, $B_{ii} \in \mathbb{R}^{n_{x_i} \times n_{u_i}}$, $A_{ij} \in \mathbb{R}^{n_{x_i} \times n_{x_j}}$; ϵ^1 is the logical variable related to the input-state-dependence of a link; Ω^{a} is the convex polyhedron for which the link ϵ^1 is activated; ϵ^2 is a control action; and ϵ^3 an external signal affecting the topology.

This multi-topological network description admits a reformulation into Mixed-Logical Dynamical (MLD) form (Bemporad & Morari, 1999), allowing the direct application of MPC control. To this, assume that the directed dynamical coupling of the jth system is defined over the polytope $\Omega^{\mathrm{a}}_j = \left\{ \left[x_j^{\mathsf{T}}; u_j^{\mathsf{T}} \right]^{\mathsf{T}} : S^{\mathrm{a}}_j x_j + R^{\mathrm{a}}_j u_j \le T^{\mathrm{a}}_j \right\}$, and we compute the constant $M^*_j \triangleq \max_{\Omega_j} S^{\mathrm{a}}_j x_j + R^{\mathrm{a}}_j u_j - T^{\mathrm{a}}_j$. Then, we introduce auxiliary variables z^1, z^2, z^3 for each edge of the graph, with $i, j \in \mathcal{V}$:

$$A_{ij} \epsilon^1_{ij}(k) x_j(k) = z^1_{ij}(k) \tag{20}$$

$$\epsilon^2_{ij}(k) z^1_{ij}(k) = z^2_{ij}(k) \tag{21}$$

$$\epsilon^3_{ij}(k) z^2_{ij}(k) = z^3_{ij}(k), \tag{22}$$

and the set of constraints that ensure the satisfaction of the logical conditions, and the correct definition of auxiliary variables:

$$S^{\mathrm{a}}_j x_j(k) - T^{\mathrm{a}}_j \le M^*_j (1 - \epsilon^1_{ij}(k)) \tag{23}$$

$$z^1_{ij}(k) \le M_j \epsilon^1_{ij}(k) \tag{24}$$

$$z^1_{ij}(k) \ge m_j \epsilon^1_{ij}(k) \tag{25}$$

$$z^1_{ij}(k) \le A_{ij} x_j(k) - m_j (1 - \epsilon^1_{ij}(k)) \tag{26}$$

$$z^1_{ij}(k) \ge A_{ij} x_j(k) - M_j (1 - \epsilon^1_{ij}(k)) \tag{27}$$

$$z^\ell_{ij}(k) \le M_j \epsilon^\ell_{ij}(k) \tag{28}$$

$$z^\ell_{ij}(k) \ge m_j \epsilon^\ell_{ij}(k) \tag{29}$$

$$z^\ell_{ij}(k) \le z^{\ell-1}_{ij}(k) - m_j (1 - \epsilon^\ell_{ij}(k)) \tag{30}$$

$$z^\ell_{ij}(k) \ge z^{\ell-1}_{ij}(k) - M_j (1 - \epsilon^\ell_{ij}(k)), \tag{31}$$

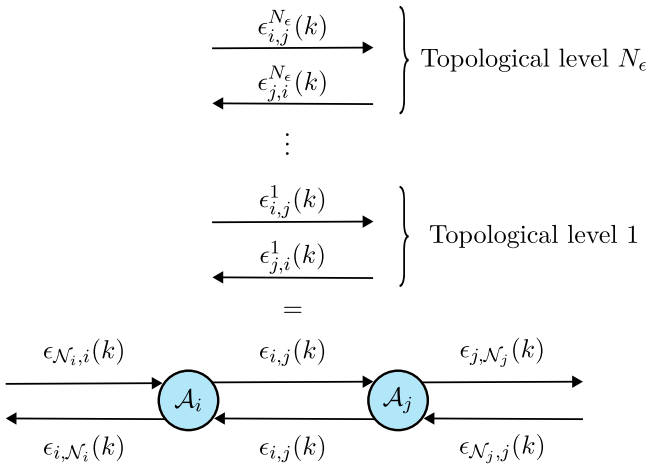


Fig. 9. General representation of the connections between agents i and j at a time-step k. The topology describing the dynamical coupling among agents at a time step k results from multiple topological levels all acting simultaneously on the network.

for $\ell = 2, 3$, allowing the definition of constraints related to the second and third variables; and $M_j = -m_j = \max_{\Omega_j} A_{ij} x_j$ are constants. The resulting system dynamics is then:

$$x_i(k+1) = A_{ii} x_i(k) + B_{ii} u_i(k) + \omega_i(k) \tag{32}$$

$$\omega_i(k) = \sum_{j \in \mathcal{N}_i} z^3_{ij}(k). \tag{33}$$

The Eqs. (23)–(33) constitute the MLD form of (16)–(19).

Remark 3. The procedure to obtain multi-topological representations presented in this section is also valid for more complex classes of systems, other than the linear ones. However, for a general nonlinear system, obtaining an MLD representation might not be possible, and more complex approaches to incorporate variable topologies into the dynamics could be required.

Remark 4. The existence of an input-state-dependent link between two areas can also be based on the configuration of both areas. In this case, the condition (19) must include variables of both areas.

4. Partitioning for predictive control

This section introduces the main ideas behind the partitioning problem for non-centralized predictive control. To this aim, we first define a specific terminology for the network components, then we introduce the metrics and the evaluation methodology to assess the quality of a partition, and finally, we present a formulation of the partitioning problem for the maximization of the performance of the control architecture. This section aims to provide the reader with a clear perspective of what partitioning optimally means, and the consequent effect on the non-centralized control architecture.

4.1. The general partitioning problem

Consider a network described by the nonlinear dynamics (3), and denoted by $\mathcal{N}$. The state, input, and output vectors are respectively $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, and $y \in \mathbb{R}^{n_y}$. The act of partitioning consists in finding a subdivision of the vectors x, u, y into a number N_C of subvectors $x_i \in \mathbb{R}^{n_{x_i}}$, $u_i \in \mathbb{R}^{n_{u_i}}$, and $y_i \in \mathbb{R}^{n_{y_i}}$ for $i = 1, \dots, N_C$, and of the respective vector fields into f_i, h_i, which describe the local subsystem dynamics:

$$C_i : \begin{cases} x_i(k+1) = f_i(x_i(k), u_i(k), w_i(k)) \\ y_i(k) = h_i(x_i(k), u_i(k)) \end{cases}, \tag{34}$$

where $w_i(k)$ represents the coupling of subsystem i with its neighboring subsystems $j \in \mathcal{N}_i$. The partition of the network is thus constituted by the set of subsystem dynamics:

$$\mathcal{P} = \{C_1, \dots, C_{N_C}\}. \tag{35}$$

Depending on the context, we call these groups C_j sets or collections of subsystems, clusters, or coalitions. This general formulation of the partitioning problem is generally too broad to be considered directly in defining a partitioning strategy. Instead, this setting has several simplified reformulations, most notably the ones reported next.

Complete non-overlapping partitioning. In (34), there is no limitation on the structure of local vectors and dynamics. However, the prevalent setting in partitioning for non-centralized predictive control is to assume that the partitioning is complete and non-overlapping, and it covers the entirety of the original dynamics. Using set notation, a complete non-overlapping partitioning $\mathcal{P}$ is such that:

$$\bigcup_{i=1}^{N_C} C_i = \mathcal{P} \quad \text{and} \tag{36}$$
$$C_i \cap C_j = \emptyset \quad \forall i, j \in \{1, \dots, N_C\} \quad \text{with}$$
$$C_i \neq 0 \ \forall i.$$

A complete non-overlapping partitioning allows the straightforward definition of local controllers and coordination protocols, making it the preferred choice in non-centralized control. Overlapping partitionings, on the other hand, are generally used to achieve performance or resilience improvements in the network.

Coupling through state dynamics. The coupling term $w_i(k)$ in (34) can, in general, comprehend both state and input interactions with neighbors, i.e. $w_i(k) = [(x_{S_j}(k))_{S_j \in \mathcal{N}_{S_i}}; (u_{S_j}(k))_{S_j \in \mathcal{N}_{S_i}}]$. However, in most settings, only state couplings are considered, yielding the vector field $f_i(x_i(k), (x_{S_j}(k))_{S_j \in \mathcal{N}_{S_i}}, u_i(k))$. This approach is the most intuitive and represents most real-world scenarios in which a local controller would be designed to steer local dynamics through the input channel u_i without directly interfering with the neighbor dynamics x_j. Moreover, it is often assumed that the output function depends only on the local state, thus taking the form $h_i(x_i(k))$. However, even if this is the most used setting in the partitioning literature, we acknowledge the presence and relevance of studies for input-coupled subsystems. We decided to treat these approaches separately in Sections 6.6, 7.7, 7.8, and 9.6. In fact, studies for input-coupled subsystems generally consider a small number of subsystems, or neglect the existence of delays in the input coupling that would introduce dynamics in the interaction among subsystems.

Fundamental subsystems. A common assumption in partitioning for non-centralized predictive control is that the network $\mathcal{N}$ in (3) admits a natural decomposition into a number N_S of atomic or fundamental subsystems that cannot be further divided for the definition of local controllers. Moreover, fundamental subsystems are coupled exclusively by state dynamics interactions, as formalized in Riccardi et al. (2025c). Therefore, the network is given as a collection of subsystems $\mathcal{N} = \{S_1, \dots, S_{N_S}\}$. In this network setting, partitioning consists in grouping the subsystems S_i into a number $N_C \leq N_S$ of bigger units C_j, i.e. using the notation (35) in defining the partition $\mathcal{P} = \{C_1, \dots, C_{N_C}\}$. Two extreme partitions are possible, one where each group is an individual subsystem, i.e. $\mathcal{P} \equiv \mathcal{N}$, $N_C = N_S$, and one that comprises the entire network, i.e. $\mathcal{P} = \{C_1\}$, $N_C = 1$.

Remark 5. The selection of fundamental subsystems does not require them to satisfy any specific control property. This choice is made because it allows us to define the fundamental subsystems for any network of systems that admits an equivalent graph representation. However, fundamental subsystems may be required to satisfy properties of interest, and for this to occur, it is sufficient to add the desired test for the property any time a new node is added to the subsystem. Moreover, additional procedures that guarantee the termination of the algorithms might be required. For example, in a network of linear systems, the selection might require the fundamental subsystems to be controllable, thus obtaining a collection of controllable fundamental subsystems, if possible.

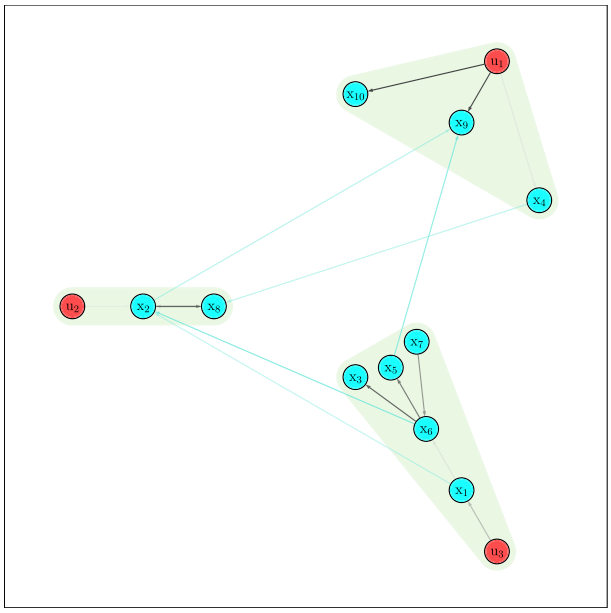


Fig. 10. A possible selection of the subsystems for the network in Example 1. The green areas indicate the subsystems and comprehend several input and state variables. The arrows that go from one subsystem to the other can be interpreted as the dynamical coupling among control agents.

Example 4. We continue Example 1 by showing a possible selection of the fundamental subsystems for that network. To this aim, we apply the algorithm for selecting fundamental subsystems defined in Riccardi et al. (2025c), which iterates over network nodes allocating them to subsystems according to coupling strengths. The resulting definition of the subsystems is given in Fig. 10. This is one definition of the subsystems, and others are possible.

Top-down and bottom-up approaches. From the discussion above, it is clear that the problem of partitioning a network can be approached from two different directions: a top-down and a bottom-up approach. In the top-down approach, a network $\mathcal{N}$ is considered a monolithic system (generally without any natural decomposition) that must be divided into smaller units. This approach is generally considered when complex nonlinear plants have to be decomposed for non-centralized control. In the bottom-up approach, instead, the problem is solved by aggregating fundamental subsystems that are given a priori, i.e. the network is assumed as $\mathcal{N} = \{S_1, \dots, S_{N_S}\}$. Both top-down and bottom-up strategies are generally valid approaches, and the preferred direction is usually dictated by the application considered.

Remark 6. When referring to a group of subsystems, we can also call it a set, cluster, or coalition. All these terms are necessarily used interchangeably throughout the survey because they all represent the same concept of a group of objects. There are subtle distinctions between the terms that will be remarked on in the specific sections. In general, the term cluster is used in the machine learning literature to indicate a group of objects that are strongly connected (Xu & Wunsch, 2005), while coalition is a term used in cooperative game theory to denote a group of players (Apt & Witzel, 2009).

4.2. Metrics and evaluation methodology

The fundamental question that each partitioning strategy in this survey tries to answer is: What is the best partition for non-centralized predictive control? This question only admits a posterior quantitative answer independently from the control strategy considered. A formal motivation for this fact is given in Section 4.3; instead, in this section, we focus on the metrics that allow us to assess the quality of a partition, and on the methodologies to do it. First of all, the best partition for a selected evaluation criterion must be defined for a specific non-centralized predictive control, i.e. Dec-MPC, DMPC, HMPC, or Coal-MPC, and w.r.t. CMPC. Throughout the section, we assume that the partition associated with CMPC is denoted by $\mathcal{P}^{\text{CMPC}}$ (this is the entire network), the one under evaluation by $\mathcal{P}^{\text{NMPC}}$, i.e. the partitioning for the application of a desired NCen-MPC strategy, and one generic partition by $\mathcal{P}^{\text{gen}}$. In this section, we first present the main metrics used to assess the quality of a partition, and then we briefly discuss the evaluation methodologies.

Metrics. In the literature, four main key performance indicators are used to assess the quality of a partition: (1) the cumulative stage cost J^{stage}; (2) the computation time J^{time}; (3) the computational cost $J^{\text{comp.}}$; and (4) the communication cost $J^{\text{comm.}}$. To validate the partition, it is necessary to simulate the system using both CMPC and NCen-MPC using the desired partitioning strategy. Then, the key performance indicators are computed as follows.

Cumulative stage cost. Assume that the stage cost for CMPC at the time step k is defined by the cost function $J(x(k), u(k-1))$. Moreover, assume a simulation horizon of N_{sim} time steps. At time step k, the optimal control problem for CMPC is solved over a horizon N, yielding a solution control sequence $\tilde{u}^*_{\text{CMPC}}(k)$, of which the first element $u^*_{\text{CMPC}}(k)$ is applied to the system, providing the next step value for the state $x_{\text{CMPC}}(k)$. Consequently, the cumulative stage cost for CMPC is

$$J^{\text{stage}}(\mathcal{P}^{\text{CMPC}}) = \sum_{k=1}^{N_{\text{sim}}} J(x_{\text{CMPC}}(k), u^*_{\text{CMPC}}(k-1)). \tag{37}$$

The cumulative stage cost for the non-centralized strategy and a selected partitioning $\mathcal{P}^{\text{NMPC}}$ is obtained similarly. However, in this case, a number $N_C = |\mathcal{P}^{\text{NMPC}}|$ of local problems is solved in parallel, providing local solutions $u^*_{\text{NMPC},i}(k)$ for $i = 1, \dots, N_C$. Then, a global vector $u^*_{\text{NMPC}}(k)$ is obtained by grouping local solutions, and is applied to the plant to compute the global state transition $x_{\text{NMPC}}(k)$. This procedure allows the computation of the cumulative stage cost for NMPC, i.e. $J^{\text{stage}}(\mathcal{P}^{\text{NMPC}})$, as done in (37) but using the non-centralized vectors. In general, for cost minimization it holds that $J^{\text{stage}}(\mathcal{P}^{\text{CMPC}}) \leq J^{\text{stage}}(\mathcal{P}^{\text{NMPC}})$. There are exceptions if the dynamics is nonlinear and the solution is obtained for a linearized version around an operating point, or if the network is subject to external uncertain signals. However, the centralized solution of the optimization problem is the reference to assess the optimality of the selected partition (and of the partitioning methodology) for a given non-centralized strategy.

An approach to compare the cumulative stage cost of different architectures consists in normalizing these results such that, for a given partition $\mathcal{P}^{\text{gen}}$ under evaluation, the normalized cumulative stage cost is given by:

$$J^{\text{stage}}_{\text{norm.}}(\mathcal{P}^{\text{gen}}) = \frac{J^{\text{stage}}(\mathcal{P}^{\text{gen}})}{J^{\text{stage}}(\mathcal{P}^{\text{CMPC}})}. \tag{38}$$

It holds that $J^{\text{stage}}_{\text{norm.}}(\mathcal{P}^{\text{CMPC}}) = 1$, and in general $J^{\text{stage}}_{\text{norm.}}(\mathcal{P}^{\text{gen}}) \geq 1$, so various partitions can be evaluated easily according to a metric that is valid across all possible strategies and applications.

Computation time. This metric is straightforward to obtain. It is sufficient to measure the execution time in seconds necessary to execute the simulation over a horizon N_{sim}. For CMPC, one CPU core is used to execute this task,[3] and the time in seconds to perform the simulation constitutes the computation time cost $J^{\text{time}}(\mathcal{P}^{\text{CMPC}})$. For NMPC, local optimization problems are solved in parallel at each time step, which requires N_C CPUs[4] The time required for this parallel execution constitutes the computation time cost for NMPC, i.e. $J^{\text{time}}(\mathcal{P}^{\text{NMPC}})$. For any well-designed non-centralized strategy and good choice of partition, it holds that $J^{\text{time}}(\mathcal{P}^{\text{NMPC}}) < J^{\text{time}}(\mathcal{P}^{\text{CMPC}})$. The gain in computation time is often one of the main reasons for deploying a non-centralized strategy. In fact, centralized computations may be prohibitive in several settings. For a partition $\mathcal{P}^{\text{gen}}$ under evaluation, the normalized version of the computation time is:

$$J^{\text{time}}_{\text{norm.}}(\mathcal{P}^{\text{gen}}) = \frac{J^{\text{time}}(\mathcal{P}^{\text{gen}})}{J^{\text{time}}(\mathcal{P}^{\text{CMPC}})}. \tag{39}$$

where $J^{\text{time}}_{\text{norm.}}(\mathcal{P}^{\text{CMPC}}) = 1$, and for a well-designed non-centralized strategy and partition $J^{\text{time}}_{\text{norm.}}(\mathcal{P}^{\text{gen}}) < 1$.

Computation cost. We discussed how, to assess the computation time in non-centralized control, it is necessary to deploy the strategy in parallel, or alternatively, perform a simulation replicating such a situation. The computation cost is a metric that quantifies the cost associated with the usage of CPUs for these parallel operations, and was introduced in Riccardi et al. (2024c, 2025c) for the evaluation of different partitions of the same network in DMPC. The best way to do so is to look at the CPU usage time, which translates immediately into power and monetary requirements once a specific technology is selected. Consequently, the unit measure of the computation cost is [core · seconds], i.e. how much CPU time in parallel is required to perform the distributed computation. For a generic predictive control strategy, being it centralized or non-centralized, the computation cost is thus assessed by computing for the simulation horizon N_{sim} the sum over the number of CPUs of the active CPUs usage time for that time step, which for a CPU i we denote by $\tau_i(k)$. If we assume that, in the non-centralized control strategy considered, one CPU is available for each agent in the partition $\mathcal{P}^{\text{NMPC}}$, then it holds that $N_{\text{CPU}} = N_C$, and the computation cost can be written as:

$$J^{\text{comp.}}(\mathcal{P}^{\text{NMPC}}) = \sum_{k=1}^{N_{\text{sim}}} \sum_{i=1}^{N_C} \tau_i(k). \tag{40}$$

It is possible to simplify this expression further if we assume that at each time step N_{sim} all local controllers will wait and idle for the slowest controller to obtain its result without performing any operation. Then, the computation cost can be written as $J^{\text{comp.}}(\mathcal{P}^{\text{NMPC}}) = \sum_{k=1}^{N_{\text{sim}}} N_C \tau^{\text{slowest}}(k)$. For both definitions of $J^{\text{comp.}}$, the normalized version of the metric for a generic partition $\mathcal{P}^{\text{gen}}$ is given by:

$$J^{\text{comp.}}_{\text{norm.}}(\mathcal{P}^{\text{gen}}) = \frac{J^{\text{comp.}}(\mathcal{P}^{\text{gen}})}{J^{\text{comp.}}(\mathcal{P}^{\text{CMPC}})}, \tag{41}$$

where $J^{\text{comp.}}_{\text{norm.}}(\mathcal{P}^{\text{CMPC}}) = 1$. In general $J^{\text{comp.}}_{\text{norm.}}(\mathcal{P}^{\text{gen}}) > 1$, but very efficient strategies can also achieve $J^{\text{comp.}}_{\text{norm.}}(\mathcal{P}^{\text{gen}}) < 1$.

Remark 7. In literature, to the authors' best knowledge, the only a priori assessment of the computational cost associated with a specific non-centralized predictive control strategy has been performed in Arastou et al. (2025). However, in that work, the determination is rather

[3] In some cases, parallel computing can also be used for CMPC. An example is when the network is constituted by hybrid systems. In this case, the MPC problem requires mixed-integer optimization, for which parallel execution algorithms are available. In such cases, instead of using one CPU for CMPC, it is possible to use any available number, given that each set of subsystems in the non-centralized strategy has such CPUs available at each time step.

[4] The analysis of the computation time can be easily extended to the case in which the number of CPUs is time-varying, i.e. for $N_C(k)$. This case occurs either when there is a time-varying partitioning $\mathcal{P}(k)$, or when the computational resources can change over time. Such extension also applies to other performance indicators.

qualitative since it is performed through a Big-O analysis of the computational complexity of the algorithm for non-centralized predictive control. In practice, such an approach cannot always establish which is better among algorithms with the same Big-O complexity, as in iterative schemes.

Communication cost. The communication cost assesses the impact of information transmission in different non-centralized control architectures. In its original formulation, see e.g. Barreiro-Gomez et al. (2019), Fele et al. (2017), Maestre and Ishii (2017) and Masero et al. (2020a) among others, the communication cost is a function of the information topology defining how coalitions in a network share information to achieve a coordinated control action. Therefore, to the non-centralized control architecture an information graph $\mathcal{G}_{\text{info}}^{\text{NMPC}} = (\mathcal{V}_{\text{info}}^{\text{NMPC}}, \mathcal{E}_{\text{info}}^{\text{NMPC}})$ is assigned, where the set of the nodes is constituted by the coalitions in the network, and the set of the edges by the active communication links. Then, to each link $\epsilon_{ij} \in \mathcal{E}_{\text{info}}^{\text{NMPC}}$ a cost is assigned, i.e. $\nu(\epsilon_{ij})$, and the communication cost is therefore computed as:

$$J^{\text{comm}}(\mathcal{P}^{\text{NMPC}}) = \sum_{\epsilon_{ij} \in \mathcal{E}_{\text{info}}^{\text{NMPC}}} \nu(\epsilon_{ij}). \tag{42}$$

This formulation of the communication cost has been used consistently in deriving coalitional control strategies, leading to partitions of the network minimizing the information sharing. The communication cost of CMPC is obtained by considering the cost associated with each possible active link in the network. The value of the cost of communication can be quantified using distance-based criteria, or the operational costs of the lines. Additionally, we stress that this approach in defining the communication cost can be used to obtain a partition, i.e. it is available a priori since it is a pure topological metric, whereas the other costs introduced before are only available a posteriori after the simulation.

While this formulation of the communication cost is direct and straightforward, it can be insufficient to establish the cost associated with iterative non-centralized control strategies. In fact, if the coordination protocol relies on the iterative sharing of information among agents to achieve an agreement about the control action to deploy, then a static topological metric can only be used to quantify the maximum amount of information shared once the maximum number of iterations of the coordination protocol is given. Posterior measurement of the true amount of information shared is, therefore, a more precise way to assess communication cost in this case. For example, assume that for an NMPC iterative strategy with information topology $\mathcal{G}_{\text{info}}^{\text{NMPC}} = (\mathcal{V}_{\text{info}}^{\text{NMPC}}, \mathcal{E}_{\text{info}}^{\text{NMPC}})$, at each time step k several iterations $N_{\text{iter}}(k)$, and at every iteration a sequence of state-input predictions of length N_{seq} is shared among the controllers. Then, for a simulation horizon N_{sim}, and assuming that each state and input variables vectors have an information transmission cost $\nu(x_i)$, $\nu(u_i)$, $i \in \mathcal{V}_{\text{info}}^{\text{NMPC}}$, then the communication cost can be defined as:

$$J^{\text{comm}}(\mathcal{P}^{\text{NMPC}}) = \sum_{k=1}^{N_{\text{sim}}} N_{\text{iter}}(k) \sum_{i \in \mathcal{V}_{\text{info}}^{\text{NMPC}}} \sum_{j \in \mathcal{N}_i} N_{\text{seq}}(\nu(x_i) + \nu(u_i)), \tag{43}$$

where the cost ν associated with the information transmission can then be directly translated into network operation or economic requirements. The CMPC strategy does not need any iteration; only variables at the current time step are shared. Therefore, its communication cost is:

$$J^{\text{comm}}(\mathcal{P}^{\text{CMPC}}) = N_{\text{sim}} \sum_{i \in \mathcal{V}_{\text{info}}^{\text{CMPC}}} \nu(x_i) + \nu(u_i). \tag{44}$$

For both formulations of the communication cost, a normalization assessment is possible. Therefore, for a given partition $\mathcal{P}^{\text{gen}}$ associated with a non-centralized MPC strategy, its normalized version is:

$$J_{\text{norm}}^{\text{comm}}(\mathcal{P}^{\text{gen}}) = \frac{J^{\text{comm}}(\mathcal{P}^{\text{gen}})}{J^{\text{comm}}(\mathcal{P}^{\text{CMPC}})}, \tag{45}$$

where $J_{\text{norm}}^{\text{comm}}(\mathcal{P}^{\text{CMPC}}) = 1$, for decentralized MPC or non-iterative strategies usually holds $J_{\text{norm}}^{\text{comm}}(\mathcal{P}^{\text{gen}}) \leq 1$, while for iterative strategies $J_{\text{norm}}^{\text{comm}}(\mathcal{P}^{\text{gen}}) \geq 1$.

Evaluation methodology. From the above discussion about metrics, it is clear that assessing the quality of a partition is mainly a task performed after a simulation or experiment is completed. This fundamental fact, i.e. the impossibility of establishing the best partitioning prior to the deployment of the strategy, is one of the main limiting factors in developing partitioning strategies for non-centralized predictive control. In fact, once a partition is selected, computationally intensive simulations involving often large (in number or size) optimization problems have to be performed. Once the metrics of interest are selected for a specific problem and control strategy, the only effective way to determine the best partition is by complete enumeration, see e.g. Atam and Kerrigan (2021). However, enumerating and testing all possible partitioning quickly becomes intractable once the number of subsystems grows by more than a few units, due to a combinatorial explosion in the number of possible partitions. Therefore, most partitioning strategies have either developed paradigms for the topological a priori evaluation of partitions, or approached the problem by maximizing the immediate gain of a performance criterion by iterative exchange of agents. A definitive statement about what is the best approach cannot be formulated yet with the current literature, which leaves open many directions for future research. In practice, there might not even be a single partition minimizing simultaneously all four indicators J^{stage}, J^{time}, $J^{\text{comp.}}$, and $J^{\text{comm.}}$. Therefore, the desired partition should be selected according to control requirements among the most promising ones.

Example 5. We show how to apply the evaluation methodology proposed in this section by comparing some of the partitioning techniques and the resulting control architectures that we found in the literature. The studies we consider report at least the cost associated with running a CMPC implementation of the controller, the one related to the NCen-MPC controller, and the respective computation times.

The results of the comparison are reported in Table 3. We considered techniques across all the partitioning classes and for different applications. The aim of this comparison is to show how different partitioning approaches affect the performance of the control strategy and the respective computation times. The values we report are in percentages. In some works, data is available for CMPC, a standard NCen-MPC technique, such as Dec-MPC or traditional DMPC, and for the NCen-MPC approach based on the partitioning method proposed. Regarding the metrics, for $J_{\text{norm}}^{\text{stage}}$, a value smaller than 1 means that the NCen-MPC based on partitioning is performing better than the reference controller, and worse otherwise. The same considerations applies to $J_{\text{norm}}^{\text{time}}$. We stress the fact that this comparison is for the sole purpose of showing what can be the effect of partitioning in NCen-MPC, and cannot be used directly as a guide to compare techniques, because they have been applied to different systems, control architectures, and with different tunings of the parameters. A fair assessment of the control performance that can be obtained with different partitioning techniques requires a dedicated study. Instead, what is interesting to notice is that NCen-MPC controllers based on partitioning often outperform more traditional DMPC or Dec-MPC approaches, and can sometimes outperform CMPC when nonlinear systems are considered.

Unfortunately, since a methodological assessment of the quality of the partition was not present in the literature, information about the computation and the communication cost of almost all the methods is not available. We suggest reporting such metrics for comparisons in future works.

4.3. Optimal partition for performance maximization

An agent $\mathcal{A}_i$ in the network is a structure with autonomy constituted by a group of subsystems $\mathcal{C}_i$, a local controller $\mathcal{K}_i$, and further devices allowing communication with other agents, or other digital features, such as the execution of algorithmic procedures.

Table 3
Comparison of different NCen-MPC control simulations for different partitioning techniques. The tables report the reference control technique w.r.t. which the metrics $J^{\text{stage}}_{\text{norm}}$ (control performance), $J^{\text{time}}_{\text{norm}}$ (computation time) are computed. A value smaller than 1 means that the NCen-MPC based on partitioning is performing better than the reference controller, and worse otherwise.

Reference study	Control and partitioning technique	Application	Reference controller	$J^{\text{stage}}_{\text{norm}}$	$J^{\text{time}}_{\text{norm}}$
Riccardi et al. (2025c)	DMPC-ADMM: based on partition index (MIQP)	Random network of hybrid dynamical systems	CMPC	1.0020	0.1660
			DMPC	0.8925	4.3971
Riccardi et al. (2025c)	DMPC-ADMM: based on partition index (Algorithm)	Random network of hybrid dynamical systems	CMPC	1.0109	0.1346
			DMPC	0.9001	3.5659
Riccardi et al. (2024c)	DMPC-ADMM: based on partition index (Genetic Algorithm)	The EEA-ENB	CMPC	1.0441	0.0364
			DMPC	1.0413	0.6548
Masero et al. (2022)	Coalition control: market-based	ACUREX plant 100 loops	CMPC	0.9320	0.1540
			Dec-MPC	0.9939	11.315
Masero, Frejo et al. (2021)	Coalition control: pairwise clustering	ACUREX plant 100 loops	CMPC	0.9340	0.0606
			Dec-MPC	0.9960	4.4565
Ocampo-Martinez et al. (2011)	Decentralized MPC: graph-partitioning-based ordering algorithm	Barcelona drinking water transport network	CMPC	1.1159	0.4698
Ocampo-Martinez et al. (2012)	Decentralized MPC: nested ϵ-decomposition	Barcelona drinking water transport network	CMPC	1.0040	0.7030
He and Li (2023)	DMPC: multiway spectral community detection algorithm	Reactor-separator process	CMPC	1.0139	0.0486
			DMPC	0.9677	0.9559
Chanfreut et al. (2021a)	Coalitional MPC: hierarchical formulation	Freeway traffic control	CMPC	1.0170	0.3860

The problem of partitioning consists of finding an allocation of the agents of the network into groups such that a set of specifications is satisfied. Different criteria, including geographical distribution, communication and coordination effort, operational constraints, security and privacy guarantees, and design choices, can guide the selection of these groups. Often, these criteria are application-dependent and, in almost all cases, are related to the control strategy to deploy. Consequently, there is no common rationale underlying all the different partitioning approaches. However, when the partitioning problem is considered in the context of non-centralized predictive control, it assumes a more precise connotation, and an optimal version can be formulated.

Assume to have a network with $N_{\mathcal{A}}$ agents, i.e. a collection $\mathcal{N} = \{\mathcal{A}_1, \dots, \mathcal{A}_{N_{\mathcal{A}}}\}$. A set C_i of N_{C_i} agents is defined as $C_i = \{\mathcal{A}_{i,1}, \dots, \mathcal{A}_{i,N_{C_i}}\}$. We introduce a matrix of binary variables $\delta \in \mathbb{M}_{N_{\mathcal{A}}}(0,1)$[5] s.t. $\delta_{ij} = 1 \Leftrightarrow \mathcal{A}_i \in C_j$. In general, we can assume δ to be time-dependent, i.e. $\delta(k)$, but time-dependence is omitted in the following for simplicity, and only used when essentially required. For a given choice δ, we denote a partition of network $\mathcal{N}$ into $N_{C(\delta)}$ sets of agents by $\mathcal{P}(\delta) = \{C_1, \dots, C_{N_{C(\delta)}}\}$. Now we consider the control performance of the network that is measured through a cost function $J(x,u,\delta)$, where x is the state of the network, u is the applied control action, and δ is the selected partitioning, a set of binary decision variables. Once the non-centralized predictive control strategy is selected, the cost J is minimized iteratively at each time step over a selected horizon N. For this, we use the vector notation $\tilde{x}_k = [x^\intercal(1|k), \dots, x^\intercal(N|k)]^\intercal$, $\tilde{u}_k = [u^\intercal(0|k), \dots, u^\intercal(N-1|k)]^\intercal$, $\tilde{\delta}_k = [\delta^\intercal(0|k), \dots, \delta^\intercal(N-1|k)]^\intercal$ to define state and input sequences over the horizon N. The global control problem is then defined as:

$$\begin{aligned}\min_{\tilde{x}_k,\tilde{u}_k,\tilde{\delta}_k} J(\tilde{x}_k,\tilde{u}_k,\tilde{\delta}_k) = &\sum_{i=1}^{N-1} J_{\text{s}}(x(i|k),u(i-1|k),\delta(i-1|k)) \\ &+ J_{\text{f}}(x(N|k),u(N-1|k),\delta(N-1|k))\end{aligned} \tag{46}$$
$$\begin{aligned}\text{s.t.}\quad & x(k+1) = f(x(k),u(k)) \\ & x(0|k) = x(k) \\ & g(\tilde{x}_k,\tilde{u}_k,\tilde{\delta}_k) \le 0,\end{aligned}$$

where J_{s} is the stage cost, J_{f} the terminal cost, and g a set of inequality constraints. This formulation of the optimal partitioning problem assumes that it is possible to simultaneously select the variables in matrix δ, and perform the steps to deploy the non-centralized control strategy. Conceptually, this contemporaneous optimization is not always possible for non-centralized control, especially if communication and coordination protocols are involved, i.e. in all cases except for purely decentralized MPC. This limitation can be overcome with a nested reformulation of (46). Specifically, the outer level is an integer optimization problem for the selection of δ, and the inner level is associated with the solution of the non-centralized control problem:

$$\begin{aligned}\min_{\tilde{\delta}_k}\quad & J^*(\tilde{\delta}_k) \\ \text{s.t.}\quad & g_{\text{out}}(\tilde{\delta}_k) \le 0 \\ & J^*(\tilde{\delta}_k) = \min_{\tilde{x}_k,\tilde{u}_k} \sum_{i=1}^{N-1} J_{\text{s}}(x(i|k),u(i-1|k))|_{\tilde{\delta}_k} \\ & \qquad + J_{\text{f}}(x(N|k),u(N-1|k))|_{\tilde{\delta}_k} \\ & \text{s.t.}\quad x(k+1) = f(x(k),u(k)) \\ & \qquad x(0|k) = x(k) \\ & \qquad g_{\text{in}}(\tilde{x}_k,\tilde{u}_k) \le 0,\end{aligned} \tag{47}$$

where, at the inner level, algorithmic procedures that ensure coordination among the agents might be present. In this formulation we assumed that the inequalities g in (46) can be split in an outer g_{out} and inner g_{in} sets depending on variables $\tilde{\delta}_k$, and $(\tilde{x}_k,\tilde{u}_k)$ respectively. This assumption usually holds since once variables $\tilde{\delta}_k$ are fixed, they do not affect further the non-centralized control strategy. Moreover, the set of constraints g_{out} can be used to impose desired properties on the partitioning. One common choice is to assume that sets C_i are non-overlapping, which can be codified with the constraints

$$\forall i \quad \sum_{j=1}^{N_{\mathcal{A}}} \delta_{ij} = 1. \tag{48}$$

[5] The class of square binary matrices of dimension $N_{\mathcal{A}}$.

The complexity of the nested optimization problem (47) is NP-hard due to the outer mixed-integer layer. Moreover, from an implementation perspective, the time requirements to find the optimal partitioning and the optimal control action with this approach can quickly become prohibitive with a growing number of agents because, for each choice of $\tilde{\delta}$, the inner non-centralized predictive control strategy might be required to perform many iterative steps involving optimization.

Remark 8. Optimal partitioning is intended for performance, but partitioning can be done according to other criteria, for which the optimal solution can be different. See Section 4.2 for a list of common metrics that can be used.

4.4. Solution methodologies

Partitioning approaches in current literature usually do not consider the level of complexity of the problem formulation (47). Instead, simplified formulations, often application-oriented, are considered. These solution approaches can be broadly categorized into the following four methodologies:

- *Static partitioning*: this is the case in which the selection of δ is made prior to the deployment of the non-centralized strategy, and the partitioning $\mathcal{P}(\delta)$ is fixed at all instants. Most approaches follow this logic due to its simplicity and the fact that the partitioning can be computed offline. The disadvantage is that changes in the network's topology cannot be compensated for with this method, making it a viable option only for stationary networks.
- *Event driven partitioning*: it is the first extension of static partitioning. When a topological change is detected, a new network partitioning is deployed. This strategy is reactive since network alterations can be detected, but no assumptions or predictions about their future behavior are made. Suppose the number of possible different topologies of the network is known a priori. In that case, all the associated partitionings of the network can be computed offline and only deployed when necessary. In other cases, the new partitioning is computed as soon as the topological change is detected, implying that the partitioning method is fast enough to be executed between two distinct MPC computations. For large networks, this is not usually suitable through optimization-based approaches. Therefore, algorithmic solutions can be considered to perform local adjustments to partitioning in the neighborhood of the topological change. Also, tabular methods can be implemented to track the topology-partitioning couples, thus avoiding re-computations in known situations.
- *Fixed partitioning over the prediction horizon*: in this case, it is assumed that the topological changes that will occur over the network during the prediction horizon are known at the current time step, either accurately or with some uncertainty. Consequently, before the start of the optimization process in the MPC, a fixed sequence $\tilde{\delta}$ can be established, and the non-centralized MPC is deployed knowing all the changes in topology and partitioning during the prediction horizon. A limited number of techniques of this type are currently available in the literature.
- *Time-varying partitioning*: this is the most complex case, where a potentially different network partitioning is allowed for each time step. In this way, all possible input-state-dependent topological changes that will occur in the network according to the available prediction model can be compensated, and uncertain topological changes might be accounted for using robustness arguments. This approach is also the only one that might guarantee the stability of the resulting non-centralized predictive control architecture under predictable topological changes. In current literature, no work is present in this category, and future research might consider addressing this problem.

Formally speaking, the last two approaches assume that a *predictive partitioning* of the network can be implemented for the NCen-MPC strategy developed. Such partitioning can assume the network topology to be static, or to change according to known rules or dynamical models. In the first case, the predictive partitioning is performed purely to improve the NCen-MPC approach. For the other two cases, there is no known approach in the literature, making predictive partitioning using models of the network topology dynamics an open problem.

We conclude this section by showing two examples of how to obtain the partition of two networks with different structures in Example 6, and of how to perform the posterior assessment of the performance of an NCen-MPC strategy applied to different partitionings of the same network in Example 7.

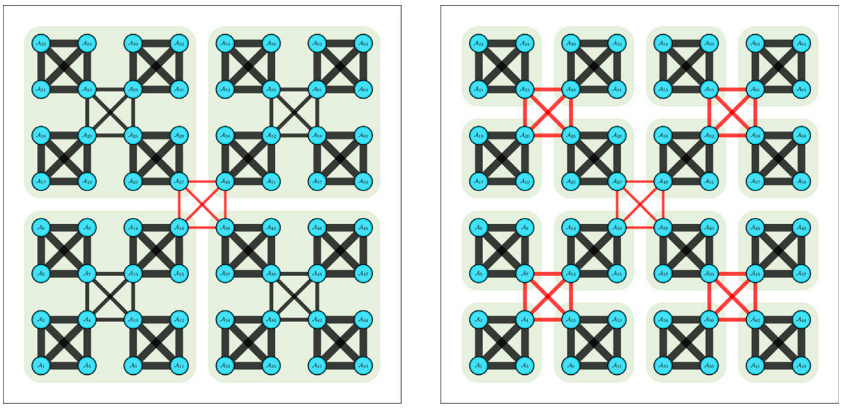
Example 6. We continue the examples started in Example 2 by showing possible partitions of the modular and random networks.

We start by considering the modular network with 64 agents, and we apply the optimization-based partitioning technique developed in Riccardi et al. (2025c). This methodology returns different optimal partitions according to a selected value for the granularity parameter, which balances coupling strengths with the size of the resulting sets of agents. Applying this partitioning methodology to the modular network returns four different partitions: the one constituted by individual agents, two partitions aggregating groups of four agents according to their modules, and the grand coalition accounting for all the agents. The examples of the two intermediate partitions are shown in Fig. 11. We also show the application of partitioning procedures defined in Riccardi et al. (2025c) to the random network with 50 agents. The use of an algorithmic approach here is advised because the previously deployed optimization-based strategy has a slow convergence rate, which is a consequence of the NP-hard nature of the problem. The algorithmic approach is instead known to have a computational complexity of at most $O(n^4)$, where n is the number of nodes of the graph, after which improvements in the partitioning quality are usually marginal, and it can be potentially optimized and parallelized as commonly done in clustering procedures (Xu & Wunsch, 2005). However, which method provides the best partitions cannot be established a priori, and the results should be validated through control experiments, which we show in the next example. Two different network partitions, one obtained through the optimization-based approach, the other through the algorithmic approach, are shown in Fig. 12.

Example 7. For this example, we consider again the random network with 50 agents. We further assume that each agent $\mathcal{A}_i$ controls a subsystem with hybrid dynamics, defined as:

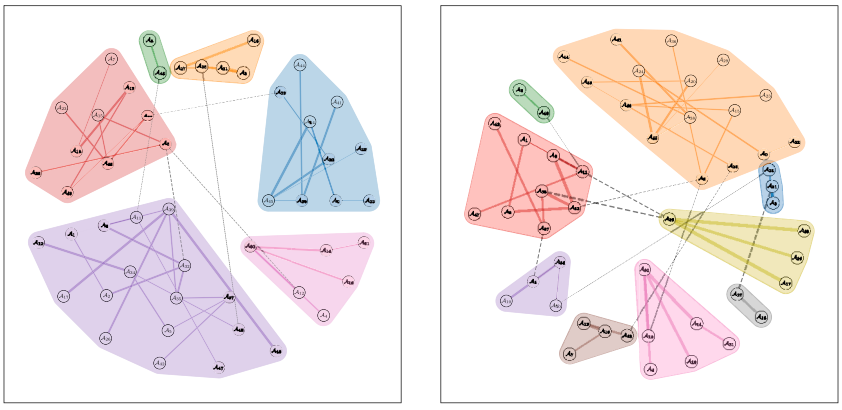
$$x_i(k+1) = 0.5x_i(k) + u_i(k) + \sum_{j\in\mathcal{N}_i} w_{i,j}x_j(k) \quad \text{if} \quad x_i(k) \geq 0$$
$$x_i(k+1) = -0.5x_i(k) + u_i(k) + \sum_{j\in\mathcal{N}_i} w_{i,j}x_j(k) \quad \text{if} \quad x_i(k) < 0.$$

Thus, subsystem $\mathcal{S}_i$ is coupled through state interactions to its neighboring subsystems $\mathcal{S}_j$ with $j \in \mathcal{N}_i$, and is subject to local constraints $u_i \in [-0.5; 0.5]$, $x_j \in [-0.9; 0.9]$ $\forall i, j$, but not to coupling constraints or objectives. The dynamical coupling occurs through the weights $w_{i,j}$, which define the topology of the network and are reported in Table 2. We want to deploy a DMPC strategy based on the alternating-direction method of multipliers (ADMM). We use hybrid dynamics because these are nonlinear systems, for which the effect of partitioning on pure network control performance is evident. Additional technical details about the case study are in Riccardi et al. (2025c). Here we focus on the results of control simulations to show how the metrics and the evaluation methodology developed in Section 4.2 can be used to assess the quality of the partitions, and to select the most appropriate partitioning strategy for the considered application. To this end, we compare CMPC and the respective coalition denoted by $\mathcal{P}^{\text{CMPC}}$, which is made by all agents; conventional DMPC, where each agent



(a) (b)

Fig. 11. Two different partitions for the modular network. The green areas represent the control agents, the black links are the interactions inside the same control agent, while the links in red represent the interactions among the control agents. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Optimization-based partition. (b) Algorithmic partition.

Fig. 12. Two possible different partitions of the random network for selected levels of the granularity parameter obtained with different strategies. These partitions are obtained with the scope of minimizing the strength of the interaction among control agents in different sets, while maximizing the interaction among control agents in the same set. While these two partitions appear to be very similar, the effects they have on network control can be quite different, as shown in Example 7.

Table 4
Comparison of DMPC-ADMM performance applied to a random network of hybrid systems for different partitioning strategies. In bold we have reported the values that show the strengths of the techniques. For example, optimization-based partitioning with DMPC $\mathcal{P}^{\text{Opt}}$ has a loss in optimality w.r.t. CMPC of the 0.24%. Instead, the partition obtained through the algorithm $\mathcal{P}_1^{\text{Alg}}$ has a loss in performance of 1.2%, while being the least expensive from a computational perspective, which is represented with a reference value of 1.00. This partition is also the fastest in terms of computation speed w.r.t. traditional DMPC, with a loss in this regard limited to the 75%.

Partition	Cores	Cost fun. value	Opt. loss %	Comp. time [s]	Comp. time ratio	Core seconds [s]	Core seconds ratio
$\mathcal{P}^{\text{CMPC}}$	1	6899	0.00	2628	26.48	2628	1.37
$\mathcal{P}^{\text{ADMM}}$	50	7749	12.31	99	1.00	4960	2.59
$\mathcal{P}^{\text{Opt}}$	6	6916	**0.24**	436	4.39	2617	**1.36**
$\mathcal{P}_1^{\text{Alg}}$	11	6982	**1.20**	173	**1.75**	1913	**1.00**
$\mathcal{P}_2^{\text{Alg}}$	9	6975	1.09	353	3.56	3184	1.66
$\mathcal{P}_3^{\text{Alg}}$	5	6911	0.16	2818	28.40	14 093	7.36

acts independently, denoted by $\mathcal{P}^{\text{ADMM}}$; one of the partitions obtained using the optimization-based method $\mathcal{P}^{\text{Opt}}$ and reported in Fig. 12; and three partitions $\mathcal{P}_i^{\text{Alg}}$ obtained with the algorithmic partitioning procedure. We propose only one optimization-based partition because they produce control simulations that are generally similar w.r.t. the algorithmic approaches that have more interesting aspects to show. The results of the control simulations are reported in Table 4. The CMPC approach has the best control performance, and is used as a reference in this category, while conventional ADMM presents a noticeable gap in performance, above the 12%. However, it is the fastest control approach, more than 26 times faster than CMPC, which can be the determining factor for selecting a specific partition in many applications. On the other hand, the computational cost in terms of core seconds w.r.t. CMPC is approximately double. The optimization-based and algorithmic control approaches provide a trade-off regarding performance gain, computation time, and cost. The strategy based on $\mathcal{P}^{\text{Opt}}$ has a negligible loss in terms of optimality, while being 6 times faster than CMPC and having approximately the same computational

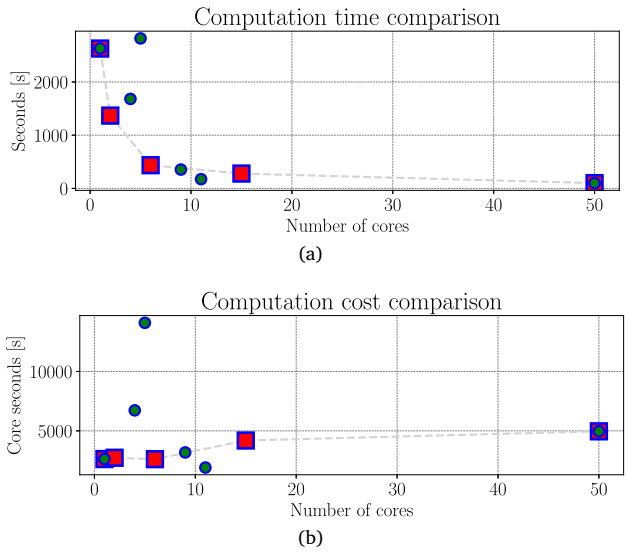


Fig. 13. Computation times and costs for solving the same NCen-MPC problem using different partitions. The data points represented with squares refer to partitions obtained through an optimization-based technique, and they approximately follow well-defined exponential or linear patterns, represented by the dashed lines. The data points using circles refer instead to the results of simulations using partitions obtained through algorithmic approaches. We see that the latter have a less clear evolution across the number of sets, which might be related to the suboptimality of algorithmic approach.

cost. Therefore, if these are a priority over speed, $\mathcal{P}^{\text{Opt}}$ is preferable w.r.t. conventional DMPC. Algorithmic partitioning approaches have mixed results. The strategy based on $\mathcal{P}_3^{\text{Alg}}$ will give the best results in terms of optimality gap, but it is also slower and more computationally expensive than CMPC; therefore, it is undoubtedly an option to discard. The approach that uses $\mathcal{P}_1^{\text{Alg}}$ has a relatively small loss in optimality, but it is also the least expensive in terms of computational cost, while retaining a good computation time. It is thus a good alternative to $\mathcal{P}^{\text{Opt}}$. The partition $\mathcal{P}_2^{\text{Alg}}$, which is the one reported in Fig. 12, offers similar results, and can also be considered. In the end, the most appropriate partition to use will depend on the requirements for the specific application, and can be selected among the listed options with a clear indication of the gains and tradeoffs. A possible way to visualize computational time and costs for different partitions, which can help guide such decisions, is reported in Fig. 13.

This illustrative example shows how posterior evaluation of operational performance for different partitions is fundamental in NCen-MPC. In particular, for the same partitioning strategy, variations in the parameters to perform the partition can lead to very different control results. This fact motivates using a solid methodological assessment of control performance under different partitions.

5. Analysis and classification of the partitioning techniques for non-centralized predictive control

In this section, we analyze and classify the partitioning techniques for NCen-MPC that we found in the literature. The analysis we perform here is oriented toward the definition of the main characteristic of each methodology, highlighting their strengths and limitations, which generally apply to all techniques belonging to that category. For a detailed technical discussion both about the general methodologies and the specific papers presented, we developed instead the sections from Sections 6 to 10.

Regarding the classification of the partitioning techniques, we propose and discuss in the following three different perspectives:

1. A categorization according to the general partitioning class, i.e. optimization-based, algorithmic, community-detection-based, game-theory-based, and heuristic.
2. A categorization in subclasses of the partitioning methods based on specific structures in the problem, or objectives to achieve through its deployment.
3. A classification according to the NCen-MPC control architecture used in the strategy.

The classification tables for the techniques in this survey are provided in Tables 5 and 7. In the first table, we collocate the works found in the literature according to class and subclass. In the second table, we classify them according to the control approach used.

5.1. Classification according to the partitioning class

Optimization-based partitioning. As introduced in Section 4, the problem of partitioning can be seen, in an abstract way, as the problem of assigning a set of objects to several given sets. This type of problem can be naturally formulated as an MIP, see e.g. Section 4.3, whose solution will provide the optimal network partitioning according to the selected metric. At the basis of this formulation, there is a binary decision variable δ_{ij} that equals 1 if the object i belongs to the set j. All partitioning methodologies based on this descriptive approach using binary variables fall into the category of optimization-based partitioning and are discussed in Section 6. When considering an optimization-based partitioning technique, it is essential to consider the fact that the associated MIP is NP-hard (Brandes et al., 2006; Karp, 1972; Sandholm et al., 1999). Consequently, their scalability is limited, and optimization-based partitioning is suitable only for relatively small problems and static network topologies. This also means that online re-partitioning of a network using optimization-based approaches is prohibitive. Approximate solutions of mixed-integer problems can be found using, e.g. the genetic algorithm (Goldberg, 1989; Srinivas & Patnaik, 1994), which does not guarantee global optimality, and still suffers from considerable computational complexity.

Algorithmic partitioning. Partitioning approaches based on algorithmic procedures are a faster and computationally less intensive alternative to optimization-based ones. The trade-off for these gains is that, unless extensive search is performed, their results are suboptimal w.r.t. the alternative optimization-based strategies, which constitutes their main disadvantage. However, for large problems or in time-varying settings, algorithmic partitioning approaches result to be the only viable option thanks to their scalability. Additionally, through algorithmic procedures, it is possible to obtain partitions according to more sophisticated requirements, such as the satisfaction of control properties, more directly and straightforwardly than through optimization-based strategies. All the approaches discussed in Section 7 fall in this category of algorithmic partitioning. However, we also stress that the works based on the community detection method reported in Section 8 are algorithmic procedures. Despite this fact, we decided to discuss community detection methods separately because: (1) it represents by itself a branch of graph and network methods, in this case applied to partitioning for NCen-MPC control; (2) a rich body of studies and approaches has been developed in partitioning for NCen-MPC control exclusively through this method; (3) in this survey, almost every community detection methodology is based on a metric called modularity. Considering these characteristics, we dedicate Section 7 to all the algorithmic methods in the literature that do not belong to the community detection approaches, and are not based on the modularity metric or its extensions. A similar consideration holds for the game-theoretic oriented partitioning approaches of Section 9. In fact, these approaches are also mainly based on algorithms; however, the fundamental presence of game-theoretic arguments in the selection of the partitions, as well as the extensive development of the coalition control methodology rooted in this technique, deserves a separate discussion in a dedicated section.

Table 5
Categorization of the partitioning techniques according to class and subclass.

Partitioning subclass	Partitioning class				
	Optimization-based	Algorithmic	Community detection	Game-theory-based	Heuristics
Unique techniques	Barreiro-Gomez et al. (2019), Núñez et al. (2015) and Xie et al. (2016)	Arastou et al. (2025), Kamelian and Salahshoor (2015), Ocampo-Martinez et al. (2012, 2011), Rocha et al. (2018) and Zheng et al. (2018); k-means: Chanfreut et al. (2023), Changqing et al. (2022), La Bella et al. (2022), Lin et al. (2020), Zhang et al. (2019) and Zhao et al. (2023)	Jogwar (2019), Jogwar and Daoutidis (2017) and Pourkargar et al. (2017)	Baldivieso Monasterios and Trodden (2021), Muros et al. (2018) and Sánchez-Amores, Martinez-Piazuelo et al. (2023)	Huanca et al. (2023), Jain et al. (2018) and Pourkargar et al. (2017)
Hierarchical		Chen et al. (2020) and Wang and Koeln (2023)	Guo et al. (2019)		
Time-varying		Kamelian and Salahshoor (2015), Rocha et al. (2018) and Wei et al. (2020)	Arastou et al. (2025) and Wang et al. (2022)	Fele et al. (2018, 2017) and Maestre and Ishii (2017)	Ananduta and Ocampo-Martinez (2021) and Liu et al. (2019)
Hierarchical time-varying	Riccardi et al. (2025c)	Chanfreut et al. (2023)	Riccardi et al. (2025c)	Chanfreut et al. (2021a), Chanfreut, Maestre, Ferramosca et al. (2022), Masero, Frejo et al. (2021), Masero et al. (2022), Masero, Maestre et al. (2021), Masero, Ruiz-Moreno et al. (2023) and Sánchez-Amores, Martinez-Piazuelo et al. (2023)	Ye et al. (2019)
Problem decomposition	Kersbergen, van den Boom et al. (2016)		Segovia et al. (2021) and Tang, Allman et al. (2018)		
Input coupling	Chanfreut, Maestre, Hatanaka et al. (2022)	Wang and Koeln (2023) and Wei et al. (2020)		Masero, Baldivieso-Monasterios et al. (2023), Masero et al. (2020b), Sánchez-Amores et al. (2022) and Sánchez-Amores, Chanfreut et al. (2023)	
Frequency-based		Tang, Pourkargar et al. (2018)	Wang et al. (2023)		
Applications	Atam and Kerrigan (2021) and Siniscalchi-Minna et al. (2020)		Guo et al. (2019), He and Li (2023), Moharir et al. (2018), Pourkargar et al. (2019), Tang et al. (2023) and Wang et al. (2022)	Chanfreut et al. (2021a), Fele et al. (2014), Maxim and Caruntu (2021, 2022) and Maxim et al. (2023, 2024)	

Despite the differences among the several partitioning approaches devised for the application of NCen-MPC considered in this survey, there are some fundamental algorithms that have been consistently used, directly or as a basis, to achieve the desired subdivision of the network. Such algorithms have usually been developed for more abstract purposes, such as graph partitioning or data clustering, and have been used across the different classes proposed in this survey. For these reasons, in Table 6 we provide an organization of the algorithmic procedures used in partitioning for NCen-MPC. We divide the algorithms into: general approaches, which are well-known strategies in the literature, and for which a reference technique is available, where their structure and pseudocode is also provided; and specialized approaches, which are techniques derived for specific problems or strictly tied to the control problem considered. For the latter case, we also provide relevant references that explain the procedure and give algorithmic details.

Community-detection-based partitioning. As mentioned in the discussion for algorithmic approaches, community detection methodologies have been developed in graph and network theory for the identification of strongly connected components of a graph for various applications (Fortunato & Hric, 2016). Among all the techniques, great attention has been devoted to community-detection-based partitioning to methods based on the maximization of the *modularity* metric (Newman, 2006). Most of the techniques in this section are conceptually based on this approach. The maximization of modularity can be either sought through the solution of an optimization problem, an NP-hard problem, or with a heuristic or greedy algorithm, where the latter approach will, in general, provide a suboptimal result. All the techniques presented here are based on the aforementioned algorithmic approaches, thus allowing for scalability and real-time applicability for time-varying partitioning. The maximization of modularity and other derived metrics will provide groups of agents that exhibit weak inter-group coupling strengths, and, potentially, strong intra-group coupling. The unproven paradigm at the basis of modularity maximization for control problems is that a partition maximizing modularity will also provide optimal NCen-MPC performance. While this statement has not been proven true or false yet, a large body of studies, presented in Section 8, has shown that partitions maximizing modularity will, in general, improve control performance w.r.t. heuristic, expert, or random partitions.

Game-theory-based partitioning. The partitioning approaches based on game-theoretic methodologies find their roots in the theory of coalition

Table 6
Algorithmic procedures used for developing partitioning approaches for NCen-MPC. The table is divided into general and specialized approaches. For general approaches, reference strategies with details about the algorithms are provided first, then the partitioning strategy for NCen-MPC is given in the application column. For specialized approaches instead, the partitioning algorithm is usually developed specifically for the problem considered, therefore details about their structure are contained in the references introducing them.

General approaches			Specialized approaches	
Algorithm	Reference technique	Application	Algorithm	Reference and application
Graph-partitioning-based ordering	Gupta (1996)	Ocampo-Martinez et al. (2011)	Based on the incidence matrix	Kamelian and Salahshoor (2015)
Nested ϵ-decomposition	Sezer and Siljak (1986)	Ocampo-Martinez et al. (2012)	Coupling degree clustering	Zheng et al. (2018)
METIS algorithm	Karypis and Kumar (1998)	La Bella et al. (2022)	For equivalent graph representations	Rocha et al. (2018)
Louvain fast unfolding algorithm	Blondel et al. (2008) and Girvan and Newman (2002)	Segovia et al. (2021), Tang, Allman et al. (2018), Tang, Pourkargar et al. (2018) and Wang et al. (2022)	Topological hierarchy decomposition algorithm	Chen et al. (2020)
k-means	Xu and Wunsch (2005)	Chanfreut et al. (2023) and Changqing et al. (2022)	Subsystems selection based on Kalman canonical form	Wei et al. (2020)
Crow search (based on k-means)	Lakshmi et al. (2018)	Zhao et al. (2023)	Maximization of partition index	Riccardi et al. (2025c)
Global fuzzy c-means	Heo and Gader (2010) and Siringoringo and Jamaluddin (2019)	Lin et al. (2020)	Based on macroscopic fundamental diagram (modularity maximization)	Guo et al. (2019)
k-shape	Paparrizos and Gravano (2015)	Zhang et al. (2019)	Game-theory-based algorithm	Fele et al. (2017), Masero et al. (2020b), Masero, Maestre et al. (2021), Maxim and Caruntu (2021, 2022) and Maxim et al. (2018, 2023)
Linkage-based Agglomerative Hierarchical Clustering (AHC)	Xu and Wunsch (2005)	Wang and Koeln (2023)	Self-organizing agents	Fele et al. (2018) and Muros et al. (2018)
Kernighan–Lin algorithm	Kernighan and Lin (1970)	Arastou et al. (2025)	Consensus-based aggregation	Baldivieso Monasterios and Trodden (2021)
Leicht and Newman algorithm (iterative bipartition)	Leicht and Newman (2008)	Jogwar and Daoutidis (2017), Moharir et al. (2018) and Pourkargar et al. (2017, 2019)	Market-based coalition formation	Masero et al. (2022), Masero, Ruiz-Moreno et al. (2023) and Sánchez-Amores, Martinez-Piazuelo et al. (2023)
Multiway spectral community detection algorithm	Zhang and Newman (2015)	Arastou et al. (2025), He and Li (2023) and Jogwar (2019)	Pairwise clustering for coalition formation	Masero, Frejo et al. (2021) and Sánchez-Amores, Martinez-Piazuelo et al. (2023)
Modified version of Barber's algorithm	Barber (2007)	Wang et al. (2023)	Inputs decomposition into private and public variables	La Bella et al. (2019), Masero, Baldivieso-Monasterios et al. (2023), Sánchez-Amores et al. (2022) and Sánchez-Amores, Chanfreut et al. (2023)
Iterative bisection with resolution parameter	Newman (2006) and Reichardt and Bornholdt (2006)	Tang et al. (2023)	Based on modal participation matrix	Jain et al. (2018)
PageRank	Brin and Page (1998) and Ishii and Tempo (2014)	Maestre and Ishii (2017) and Muros et al. (2018)	Based on wind forecast	Ye et al. (2019)
Modified graph-partitioning-based ordering	Ocampo-Martinez et al. (2011)	Chanfreut, Maestre, Ferramosca et al. (2022) and Maestre et al. (2014)	For vehicle platoons	Liu et al. (2019)
Genetic algorithm	Goldberg (1989)	Chanfreut et al. (2021a) and Riccardi et al. (2024c)	For economic dispatch	Ananduta and Ocampo-Martinez (2021)
Based on the sphere packing problem	Conway and Sloane (1988)	Christofides et al. (2013)		

formation (Ray, 2007). Agents in such networks participate in a game in which they seek their best allocation in a coalition to maximize the collective outcome, which, in this context, corresponds to the global cost function of the MPC problem. Most of the partitioning strategies developed in this field are based on algorithmic procedures; however, the prominent presence of game-theoretic techniques, and the fact that a whole body of literature has been developed about the resulting control strategy, i.e. Coal-MPC, motivate the treatment of these methodologies in a dedicated section. Game-theoretic partitioning methodologies are, in general, more complex to develop w.r.t. other algorithmic approaches, and require the clear definition of cooperative games and the associated cost functions. However, these approaches also allow for obtaining interpretable performance gains in the deployment of the Coal-MPC strategy, a point often missing in most algorithmic approaches.

Heuristic partitioning. In this class, we include all the partitioning strategies for NCen-MPC that we found in the literature based on heuristic methodologies, which have not been developed originally to be extended to other applications. While the scope and generalizability of these strategies may appear limited, they can still be highly effective in specific contexts, and may offer inspiration for developing more broadly applicable methods.

5.2. Classification according to the partitioning subclass

As it is possible to see in Table 5, there are common features shared among partitioning techniques across different general partitioning strategies.

First, we can identify hierarchical strategies, in which we collocate approaches that either have multiple aggregation levels for the resulting partition, or are developed using a partitioning layer distinguished from the control layer. All purely hierarchical approaches presented in Table 5 belong to the first category. Among these, we find works that use a hierarchy to introduce a sequential decision-making ordering into the NCen-MPC strategy, or works with multi-level partitioning approaches, generally used for partition refinement. The former approaches allow obtaining coordinated actions prioritizing the performance of the controllers at the highest level of the hierarchy, and sacrificing the others; the latter generally use purely topological metrics, thus not being directly oriented toward performance optimization.

Time-varying approaches include the techniques that allow for a reconfiguration of the network, either online during the execution of the control strategy or offline through the derivation of look-up tables. These methods are developed to react to topological changes in the network with the objective of maximizing the global operation cost. While real-time adaptability of the partition is advisable (when possible) to improve performance, the computational complexity of the partitioning problem can make it prohibitive if the network has fast dynamics. On the other hand, the offline computation of pre-defined partitions will surely allow for fast online reaction to topological changes, but on the other hand, it assumes either that it is possible to compute all these desired partitions, or there is a trade-off between performance and quality of the partition according to heuristics.

Hierarchical time-varying strategies are obtained by combining the two previous concepts. The most common setting is the following: a partitioning layer generally operates at a higher hierarchy level and a slower time scale w.r.t. a control layer. This approach has been extensively explored because the execution of a partitioning strategy cannot generally be performed in real time according to the control sampling time. Therefore, a slower time scale is used for the partitioning layer, allowing either periodic or event-driven network reconfiguration. Hierarchical time-varying strategies allow to obtain enhanced control performance, generally adapting the partitioning (reactively) w.r.t. network performance; however, two main aspects deserve some attention: (1) these are complex strategies, and therefore they require a higher level of coordination and communication w.r.t. more direct approaches (2) operating at different time scales allows for online re-partitioning, but assumes that the performance degradation during the partitioning intervals is acceptable, and eventual topological changes between re-partitioning intervals will not harm network operation.

Partitioning for input-coupled dynamics has been addressed separately because the underlying dynamics lead to strategies that present unique features, such as the definition of private and public control actions and related negotiation strategies, which are usually not considered when the dynamics present coupling through state interactions. In theory, most of the techniques defined for dynamical coupling among network subsystems can be extended to input-coupled dynamics with the necessary care. The most critical aspect for these systems is their limited applicability to real-world problems, which is also reflected in the limited amount of related studies.

Frequency-based approaches are defined based on the network's transfer functions that link input–output channels. These approaches find their roots in the MIMO decoupling approaches (Skogestad & Postlethwaite, 2001) for selecting control channels. Frequency-based approaches are generally developed for linear or linearized systems, and instead of using a direct performance assessment for partitioning, they use frequency-based performance metrics.

A range of approaches in the literature can be seen as applicative work of previously developed strategies, or as prototype techniques that have been extended later. These works can be used to develop comparative case studies for future developments.

Finally, a range of techniques has been uniquely defined in each partitioning methodology. These works do not share their direct scope with others; thus, we have placed them in a separate category. However, their features can potentially be extended to other techniques, and direct comparisons might be possible.

5.3. Classification according to the partitioning methodology

A further classification of the partitioning techniques can be provided in terms of the methodology they are designed for. Specifically, a partitioning strategy can be either developed to operate on a given structure, or to address a specific problem. This classification is provided in Fig. 4 as a coloring scheme to distinguish the methodology to which all the subclass entries belong, where mixed approaches indicate that both methodologies have been used in the same subclass. In the following, we discuss their characteristics.

Structure-based partitioning strategies leverage the presence of a structure in the topology of the network or optimization problem to obtain the partition. Generally speaking, these approaches only require information about the network connections, and can use well-known tools from network and graph theory, such as spectral clustering or k-means. One reason to use such approaches is that for some applications, knowing the dynamics of the network is not essential for the specific partitioning problem, and other factors, such as achieving a particular decomposition for ease of operation, accessibility, or maintenance of the network, must be taken into account. Additionally, structure-based approaches do not generally need any information about the dynamics of the subsystems in the network, which can be advantageous in settings where security and privacy are of main concern. In this context, pairing structure-based partitioning with Dec-MPC approaches can be advisable. In such settings, there will be no requirement for real-time data or a communication infrastructure, and the approach can work well in situations where the network does not change over time, or changes slowly and predictably. The main trade-off in such implementations will be the loss in control performance, and the adaptability of the control structure. However, it is important to stress that structure-based approaches should not be limited to static networks, because they can also be developed for time-varying networks and be used with communication-based NCen-MPC approaches. Their main drawback in this sense is that they do not generally account directly for the dynamics of the subsystems; therefore, their actual impact on the performance should be quantified a posteriori.

Goal-oriented partitioning strategies are, in a sense, oriented toward the opposite direction compared to structure-based ones. In fact, they are developed to achieve a given goal without explicitly accounting for the structure of the problem. Usually, this is a control goal, and often, performance optimization. To this, goal-oriented partitioning must have access to some form of information that can relate to the predictable behaviors of the network, such as subsystem dynamics, time-series predictions from local controllers, or the operation cost of the local optimization problems. Additionally, communication and coordination structures are required to leverage and process such information, which increases development costs and complexity; but also affects the privacy of agents and security of network operation. Additionally, goal-oriented partitioning is naturally suited to work with

time-varying networks, because it already requires real-time data about the current operation. It also follows that goal-oriented partitioning can be paired effectively with communication-based NCen-MPC, such as DMPC, HMPC, and Coal-MPC. The advantage here is generally sought in performance optimization, or to achieve particular configurations of agents for specific tasks.

From this discussion, it is clear that both partitioning methodologies are fundamental in the literature, and research in the field of MPC should keep addressing both themes.

5.4. Classification according to the control strategy

In Table 7, we categorize the works in partitioning according to the control architecture to which they have been applied. Other than the more conventional Dec-MPC, DMPC, and HMPC strategies, we report that extensive work has been performed on the Coal-MPC methodology. Instead, few studies involve nonlinear MPC strategies. Finally, we mention the presence of a few mixed control strategies that allow for switching between control architectures according to control necessities. In the following, we briefly discuss each strategy, but for a detailed discussion, we refer the reader to Fele et al. (2017), Maestre and Negenborn (2014) and Scattolini (2009).

Starting from the simplest form of NCen-MPC, we have Dec-MPC in which local controllers do not share any information with their neighbors and compute the local control actions either independently, or using some approximated or estimated information about the strength of the incoming dynamical coupling. Robustness arguments are used to ensure the stability of the network under uncoordinated operation. The biggest strength of Dec-MPC, other than the non-centralized computation of the control action, lies in the ability to preserve the privacy of local subsystems during network operation, since there is no information sharing. The main drawback is the loss of performance w.r.t. CMPC, given the conservative nature of local actions.

In the DMPC approach, information about the current state of the local subsystem, the current control action, or even the predicted state-input sequence is shared among local controllers. This communication is supported by a coordination protocol, which allows local controllers to refine the local actions to achieve superior global performance for the network. The communication and coordination strategy can be structured according to different criteria, thus producing different DMPC approaches. In linear settings, DMPC strategies can converge to near CMPC performance, which is the main advantage of DMPC. However, DMPC also has drawbacks: more expensive hardware requirements w.r.t. Dec-MPC, due to the communication infrastructure and the necessity of more advanced abilities for local controllers; complex coordination algorithms, which can also be iterative and must operate within the limits of real-time control; information sharing, which is not always guaranteed to be possible or real-time.

HMPC includes any control strategy having local controllers and a coordination layer in the form of a centralized decision maker. Such approaches are usually designed to achieve performance advantages, while allowing to overcome other technical challenges, such as model complexity reduction, multi-scale network operation, privacy preservation, or optimization of global coordination. Given the flexibility of HMPC approaches, the specific drawbacks of each technique depend on its implementation, but all approaches unquestionably come at the cost of an increased technical complexity and increased hardware requirements w.r.t. simpler NCen-MPC approaches.

The Coal-MPC strategy was born to fuse MPC with game theory in a non-centralized control setting. The result is a control strategy in which local control agents can merge into coalitions according to game-theoretic strategies to achieve superior control performance. Therefore, the Coal-MPC strategy can also be interpreted by itself as a game-theoretic-oriented partitioning strategy for distributed MPC, with dynamic allocation of local controllers into time-varying coalitions. In this regard, the Coal-MPC problem inherits the computational complexity of the general partitioning problem, or coalition formation problem, i.e. it requires the online solution of an NP-hard problem. This main drawback has been solved through different algorithmic procedures, which has led to the development of a large body of literature also discussed in this survey. The main theoretical advantage of Coal-MPC is that it allows for online dynamic partitioning with the objective of global performance optimization in a game-theoretic sense.

Regarding NLin-MPC, the above considerations have to be extended in a setting where the MPC model is nonlinear. This approach can allow for superior operational performance, but has several drawbacks, mainly: the complexity of defining an appropriate nonlinear model, the computational complexity related to nonlinear optimization, the eventual presence of local minima in the cost function, and the difficulty in ensuring stability of operation.

Mixed strategies for NCen-MPC use any combination of the previous techniques, trying to balance their strengths and limitations with online reconfiguration of the controllers' settings and (sometimes) partitions. This fact necessarily implies that such strategies have a high implementation complexity, and a combinatorial number of possible approaches at each time step, which is usually addressed through the use of heuristics.

6. Optimization-based partitioning

6.1. General techniques

In the (Núñez et al., 2015), a partitioning strategy is proposed that allows deploying a decentralized, distributed, or hierarchical MPC controller as a function of the operating conditions of the network. This is achieved by introducing an integer variable δ_{ij} for each edge ϵ_{ij} of the information graph.

Input–output decomposition of large-scale linear systems is performed in Xie et al. (2016) for the application of DMPC. The work proposes a two-stage procedure consisting of an input clustering decomposition (ICD) first, and then of an input–output pairing decomposition (IOPD). The scope of the IOPD partitioning here is to minimize the coupling effect among the subsystems, defined as:

$$J^{\text{coupling}} = \frac{\|O - \text{diag}(O_{11}, \dots, O_{MM})\|_{\text{F}}}{\|O\|_{\text{F}}}, \tag{49}$$

where the matrix O is a function of the ICD and of the stage cost of the MPC program.

6.2. Multi-objective optimization in partitioning

Network Partitioning is achieved through a multiobjective optimization program in Barreiro-Gomez et al. (2019), where, at each time step k, the following problem is considered:

$$\begin{aligned} \min_{\mathcal{P}(k)} \ & \sum_{i=1}^{4} \varphi_i \sigma_i(\mathcal{P}(k)) \\ \text{s.t.} \ & \bigcup_i C_i(k) = \mathcal{P}(k) \\ & \bigcap_i C_i(k) = \emptyset. \end{aligned} \tag{50}$$

Constraints on the sets $C_i(k)$ ensure retrieving a nonoverlapping partition $\mathcal{P}(k) = \{C_1(k), \dots, C_{N_C}(k)\}$; the weights φ_i allow prioritizing the four topological indicators σ_i. The problem is solved through the distributed approach using the Kernighan–Lin algorithm (Gupta, 1996).

Table 7
Categorization of the partitioning techniques according to the control strategy deployed.

Control approach	Partitioning class				
	Optimization-based	Algorithmic	Community detection	Game-theory-based	Heuristics
Decentralized MPC	Atam and Kerrigan (2021) and Núñez et al. (2015)	Kamelian and Salahshoor (2015), Ocampo-Martinez et al. (2012, 2011) and Wang and Koeln (2023)	Arastou et al. (2025) and Wang et al. (2022)	Baldivieso Monasterios and Trodden (2021)	Jain et al. (2018)
Distributed MPC	Barreiro-Gomez et al. (2019), Kersbergen, van den Boom et al. (2016), Núñez et al. (2015), Riccardi et al. (2025c) and Xie et al. (2016)	Arastou et al. (2025), La Bella et al. (2022), Rocha et al. (2018), Tang, Pourkargar et al. (2018), Wei et al. (2020), Zhang et al. (2019) and Zheng et al. (2018)	Arastou et al. (2025), Guo et al. (2019), Jogwar (2019), Moharir et al. (2018), Pourkargar et al. (2017, 2019), Riccardi et al. (2025c), Segovia et al. (2021), Tang, Allman et al. (2018), Tang et al. (2023) and Wang et al. (2023)	Maxim and Caruntu (2021) and Maxim et al. (2023)	Huanca et al. (2023), Liu et al. (2019) and Pourkargar et al. (2017)
Hierarchical MPC	Núñez et al. (2015) and Siniscalchi-Minna et al. (2020)	Chanfreut et al. (2023), Changqing et al. (2022), Chen et al. (2020), Lin et al. (2020), Ocampo-Martinez et al. (2012) and Zhao et al. (2023)	He and Li (2023)		Ye et al. (2019)
Coalitional MPC	Chanfreut, Maestre, Hatanaka et al. (2022)			Chanfreut et al. (2021a), Fele et al. (2018, 2017, 2014), Maestre and Ishii (2017), Masero, Baldivieso-Monasterios et al. (2023), Masero et al. (2020b), Masero, Frejo et al. (2021), Masero et al. (2022), Masero, Maestre et al. (2021), Masero, Ruiz-Moreno et al. (2023), Maxim and Caruntu (2021, 2022), Maxim et al. (2023, 2024), Muros et al. (2018), Sánchez-Amores, Chanfreut et al. (2023) and Sánchez-Amores, Martinez-Piazuelo et al. (2023)	
Nonlinear MPC		Kamelian and Salahshoor (2015) and Rocha et al. (2018)	Tang, Allman et al. (2018)		
Mixed strategies	Núñez et al. (2015)			Chanfreut, Maestre, Ferramosca et al. (2022), Maxim and Caruntu (2021) and Maxim et al. (2023)	Ananduta and Ocampo-Martinez (2021)

6.3. For optimization problem decomposition

In Kersbergen, van den Boom et al. (2016), DMPC for hybrid systems is considered. The centralized MPC problem provides MILP program, defined as:

$$\min_{z(k)} c^{\intercal}(k)z(k) \tag{51}$$
$$\text{s.t. } A(k)z(k) \leq b(k).$$

Such a program is partitioned into nonoverlapping subproblems such that constraints are decoupled, and the size and number of variables of the problems are approximately the same. Then the subproblems are solved in parallel in a DMPC fashion. The cost of the partitioning problem is:

$$J = \rho M_{\text{MAX}} - \sum_{j=1}^{n_{\text{T}_2}} \sum_{k=1}^{n_{\text{T}_2}} \sum_{i=1}^{N_C} \delta_{ji} Q_{jk} \delta_{ki}, \tag{52}$$

where n_{T_2} is the number of constraints of (51) and N_C its the number of binary variables; matrix Q represents the topology of the constraints interconnection; ρ is a tuning parameter; and M_{MAX} the maximum difference in the number of constraints in the subproblems.

6.4. Ad-hoc performance indicators

Non-centralized hierarchical control of wind farms is considered in Siniscalchi-Minna et al. (2020), where MPC is used for reference point setting at the control partition level, while conventional controllers are used for individual turbines. A weighted directed graph is constructed using the intensity of the wakes[6] as labeling (Annoni

[6] The wake-effect refers to the wind reduction and increased turbulence that downstream turbines experience due to the extraction of wind power from upstream turbines.

et al., 2018). Multiobjective integer optimization is used to maximize the wakes, minimize the distance of the turbines in the same groups, and balance the size of the groups.

6.5. Robust and stochastic optimization

Robust and stochastic methodologies for partitioning have been developed in Atam and Kerrigan (2021), where for Dec-MPC of the thermal zones of a building is considered. Thermal interactions among the zones are used to formulate a mixed-integer optimization problem for partitioning (Boulle, 2004). In the stochastic formulation, thermal interactions are replaced by their expectations, while in the robust formulation by their worst-case scenario.

6.6. Input-coupled dynamics

Binary quadratic programming (BQP) is used in Chanfreut, Maestre, Hatanaka et al. (2022) to partition input-coupled systems of the form:

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + w_i(k) \tag{53}$$
$$w_i(k) = \sum_{j\in\mathcal{N}_i} B_{ij}u_j(k).$$

The approach is based on the gradient approximation of the cost function J, as defined in De Oliveira and Camponogara (2010), which, for a topology Λ, is defined as:

$$g^{\Lambda} = \nabla J(\tilde{x},\tilde{u}) \approx g^{\text{local}} + \sum_{ij\in\Lambda} \Delta g_{ij}^{\Lambda}. \tag{54}$$

In this approximation, g^{local} are the local contributions, while Δg_{ij}^{Λ} are coupling contributions for Λ. Partitioning is then obtained through a BQP using g^{Λ} in the cost function.

6.7. Hierarchical approaches for time-varying graphs

A BQP approach potentially applicable to time-varying topologies has been developed in Riccardi et al. (2025b, 2025c), allowing to obtain partitions at different levels of aggregation through a granularity parameter α. The strategy is based on the construction of a graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ with a weighting of the edges defined by (4), and on a preliminary selection of fundamental system units (FSUs) $\mathcal{S}_i$ through the algorithm detailed in Section 8.6. The BQP aggregates FSUs into collections C_i, called composite system units (CSUs). Binary variables $\delta_{ij} = 1 \Leftrightarrow \mathcal{S}_i \in C_j$, are defined for this scope together with the weighting functions:

$$W^{\text{inter}}(\delta) = \sum_{m=1}^{N_{\text{FSU}}}\sum_{i=1}^{N_{\text{FSU}}}\sum_{\substack{j=1\\ j\neq i}}^{N_{\text{FSU}}}\sum_{\substack{l=1\\ l\neq m}}^{N_{\text{FSU}}} \delta_{i,m}\delta_{j,l}\left(|w(i,j)| + |w(j,i)|\right) \tag{55}$$

$$W^{\text{intra}}(\delta) = \sum_{m=1}^{N_{\text{FSU}}}\sum_{i=1}^{N_{\text{FSU}}}\sum_{j=1}^{N_{\text{FSU}}} \delta_{i,m}\delta_{j,m}\left(|w(i,i)| + |w(i,j)| + |w(j,i)| + |w(j,j)|\right) \tag{56}$$

$$W^{\text{size}}(\delta) = \sum_{m=1}^{N_{\text{FSU}}}\left(\sum_{i=1}^{N_{\text{FSU}}}\delta_{i,m}\right)^2.$$

These three weights represent the interaction strength within and among the collections for a given value of the parameter α that influences the level of granularity:

$$\begin{aligned}\min_{\delta}\quad & W^{\text{inter}}(\delta) - W^{\text{intra}}(\delta) + \alpha W^{\text{size}}(\delta)\\ \text{s.t.}\quad & \sum_{j=1}^{N_{\text{FSU}}}\delta_{ij} = 1 \qquad \forall i\\ & \delta_{ij}\in\{0,1\}.\end{aligned} \tag{57}$$

The constraints ensure that nonoverlapping sets constitute the resulting partitioning, and varying α allows for obtaining collections of different sizes.

7. Algorithmic partitioning

7.1. Applied to equivalent graph-based representations

One of the first contributions to graph-based partitioning for the application of non-centralized predictive control is found in Ocampo-Martinez et al. (2011). The starting point of the partitioning strategy is a graph-based representation, proposed as a control-oriented representation described by an incidence matrix (Bondy & Murty, 2008; Zecevic & Siljak, 2010). The graph is divided into non-overlapping subgraphs according to an algorithm developed starting from the graph-partitioning-based ordering algorithm (GPB) (Gupta, 1996), with various modifications and heuristics to adapt it for control of a complex system. One of the core components of the algorithm is the cut size, i.e. the number of links that belong to different subgraphs, which is an indirect measure of the desired subgraph size.

Partitioning based on nested ϵ-decomposition (Sezer & Siljak, 1986) is proposed in Ocampo-Martinez et al. (2012) for decentralized predictive control. For a linear causal system, the ϵ-decomposition works as follows. Construct a matrix M using all system variables as nodes of a graph, i.e. build the weighted adjacency matrix:

$$M = \begin{bmatrix} A & B & 0\\ 0 & 0 & 0\\ C & 0 & 0\end{bmatrix}. \tag{58}$$

Then, for a given threshold ϵ, compute the permutation matrix P that provides a new block decomposed matrix $\tilde{M} = P^{\intercal}MP$ consisting of N block such that, for the off-diagonal terms, it holds that $\tilde{M}_{ij} \leq \epsilon$. This decomposition transforms the network into N connected subgraphs where interconnections are defined by the off-diagonal terms of $\tilde{M}$ and their strength constrained by the choice of ϵ. A maximum number of $|M|$ nested ϵ-decompositions is possible for any given M. Further details and stability analysis of this decomposition are presented in Sezer and Siljak (1986).

An algorithmic approach for nonlinear systems is devised in Kamelian and Salahshoor (2015). This approach is also based on the control-oriented representation and the derivation of the incidence matrix (Bondy & Murty, 2008); however, in this case the matrix is constructed accounting for relations among system variables, where each input state and output is considered as a distinct node. A general nonlinear dynamics of the form:

$$\mathcal{S}: \begin{cases} x_{k+1} = f(x_k,u_k,w_k)\\ y_k = g(x_k,u_k,w_k)\end{cases}, \tag{59}$$

is used to construct the graph; however, this dynamics is linearized around an operating point to derive a weighting of the associated graph; specifically, the matrices (A,B,C,D) resulting from the linearization are used. The algorithmic approach starts by the centers of the clusters as the input variables. Then, a sorting procedure is used to order the state and output vertices according to their degree. A merging phase groups subgraphs based on their number of edges. The procedure is regulated by the cut size, according to Jamoom et al. (1998), but also considering the number of resulting groups.

A partitioning approach based on the strength of interaction among subsystems is proposed in Zheng et al. (2018). The approach requires subsystems to be grouped into larger virtual middle-scale subsystems, which are selected to be weakly coupled according to a condition defined in the paper. Then, a variable adjacency matrix $A(\delta) = (a)_{ij}$ function of the threshold δ is obtained as $a_{ij} = 1$ if $\|A_{ij}\| \geq \delta$, and $a_{ij} = 1$ otherwise. The algorithmic clustering approach consists of finding such δ and a permutation matrix T such that $T^{\intercal}\tilde{A}T$ is block-diagonal, and the overall system is weakly coupled. The clustering algorithm consists of gradually reducing δ from a given initial value δ_0 until the decomposition into weakly coupled middle-scale subsystems is achieved

A framework for algorithmic partitioning of nonlinear systems based on the equivalent graph representation of linearized dynamics around

an operating point is proposed in Rocha et al. (2018). In this approach, each time a re-linearization of the nonlinear dynamics is performed, the system might be re-partitioned. The partitioning algorithm proposed is based on the iterative grouping of input-state-output variables, followed by a controllability check. The algorithm does not guarantee the terminability, or that controllable groups are achievable.

7.2. Applied to flow graph representations

Algorithmic partitioning for power networks using a flow-graph representation is considered in La Bella et al. (2022). First, the power network is divided into sources for generators (a set of nodes $\mathcal{V}^{\text{source}}$), and sinks for the loads (a set of nodes $\mathcal{V}^{\text{sink}}$), thus constructing a flow graph. Then, the optimal power flow problem (Frank & Rebennack, 2016) for the best and worst case scenarios is solved The average of these two solutions allows defining the average transaction $x^*_{ij}(k)$ between sources $i \in \mathcal{V}^{\text{source}}$, and sinks $j \in \mathcal{V}^{\text{sink}}$. Then, for each $i \in \mathcal{V}^{\text{source}}$ and $j \in \mathcal{V}^{\text{sink}}$ the shortest path $\mathcal{L}_{ij}$ is defined (Dijkstra, 1959), and the value $x^*_{ij}(k)$ is assigned to all edges in $\epsilon \in \mathcal{L}_{ij}$. Finally, the weight of each edge in the network is computed by summing all the values $x^*_{ij}(k)$ of the shortest paths passing by that edge. A weighted flow graph is thus constructed. A partitioning of this graph for a given number of clusters is obtained using the k-way partitioning method minimizing the edge cut using the METIS algorithm (Karypis & Kumar, 1998). This procedure is performed at the time scale of the clustering procedure, slower than the time scale of the control process.

7.3. Using frequency-based performance indicators

The authors in Tang, Pourkargar et al. (2018) propose a new metric called Relative Time-Averaged Gain Array (RTAGA), based on the step response of the system averaged by an exponential distribution function $f(t,\tau) = (1/\tau)\cdot e^{-t/\tau}$, for a parameter τ characterizing the decay of the exponential. Then, for matrix G of transfer functions, the element $g_{ij}(1/\tau)$ is the intensity of the response y_i for a step input u_i weighted by the distribution $f(t,\tau)$ decaying at time scale τ. Accordingly, the RTAGA matrix is defined as $\Lambda(1/\tau) = G(1/\tau)\cdot G^{\intercal}(1/\tau)$. For partitioning the system using the RTGA, the authors of Tang, Pourkargar et al. (2018) rely on the input–output bipartite graph, and the weighting of each edge (u_j, y_i) representing input–output loops is given by the scalar w_{ij} defined according to the entries of $\Lambda(1/\tau)$ as:

$$w_{ij} = \begin{cases} \lambda_{ij} & 0 \le \lambda_{ij} \le 1 \\ 1/\lambda_{ij} & \lambda_{ij} > 1 \\ 0 & \lambda_{ij} < 0 \end{cases}. \tag{60}$$

Then the modularity Q of the bipartite weighted graph is defined according to Barber (2007), and modularity maximization is achieved through the Louvain fast unfolding algorithm (Blondel et al., 2008).

7.4. Using k-means

One of the most used algorithms for clustering is k-means (Xu & Wunsch, 2005). At the core of the algorithm there is the problem of organizing N objects, e.g. vectors $x \in \mathbb{R}^d$, into K subsets. This is achieved by using the definition of Euclidean distance, and an algorithm is developed to minimize the squared error between each object and the center of the clusters. The algorithm starts with an initialization of the centers of the K clusters (either random or informed). Then, each object is assigned to the nearest cluster. Accordingly, the prototype matrix, i.e. the matrix containing centroids or the means of the clustering, is updated with the given assignment. The last two steps are iterated until there is no further change in the clusters. The computational cost of the algorithm is $O(N\,K\,d)$. The k-means clustering is well developed, and parallel implementations are available (Stoffel & Belkoniene, 1999) to improve computation times. The interested reader can refer to the survey (Xu & Wunsch, 2005) for further information.

The clustering of a wind farm using k-means has been performed in Changqing et al. (2022). The article focuses on the frequency regulation of a double-fed induction generator, which is affected by both the operating conditions of the plant, and the wind orientation and strength. To improve the frequency regulation of the system, a multi-layer control approach is proposed: MPC (Afram et al., 2017) is used for frequency regulation and power output maximization, whereas k-means clustering (Vallee et al., 2011) based on wake-effect interaction is used to spatially cluster the wind turbines.

An improved version of k-means, i.e. crow search (Lakshmi et al., 2018), is used in Zhao et al. (2023) to cluster a wind farm. Crow search is used in this approach for its improved clustering accuracy and cluster stability, allowing the authors of Zhao et al. (2023) to achieve superior cluster quality w.r.t. traditional k-means. The wind farm is partitioned according to four key performance indicators, which are the power characteristic of the turbine, the smooth coefficient, the generation potential coefficient, and the anomaly coefficient (Howlader et al., 2015; Yin et al., 2022). Given this dataset, the algorithmic partitioning is performed for a given number of clusters.

An approach for clustering wind farms based on an approximate linear model of their power tracking (Chen et al., 2019; Jha, 2010) is proposed in Lin et al. (2020). Once an estimate of this transfer function is available for each turbine in the farm, Lin et al. (2020) proposes to apply a global fuzzy c-means algorithm for clustering the network (Heo & Gader, 2010; Siringoringo & Jamaluddin, 2019).

7.5. Data-driven decomposition

Partitioning in a data-driven application is discussed in Zhang et al. (2019). The scope of a data-driven approach is to capture the nonlinear dynamics that might not figure in purely model-based approaches as Ocampo-Martinez et al. (2012). Once time series data about inputs $\mathbb{U} = \{\tilde{u}_\ell\}_{\ell=1}^{n_u}$, states $\mathbb{X} = \{\tilde{x}_\ell\}_{\ell=1}^{n_x}$, and outputs $\mathbb{Y} = \{\tilde{y}_\ell\}_{\ell=1}^{n_y}$ are collected, a system model is defined as $\mathcal{S}(\mathbb{U},\mathbb{X},\mathbb{Y})$. The partitioning problem is then formulated s.t. the network is divided into k subsystems $\mathcal{S}_i$, where k is a number defined by inspection depending on the shape of the time-series data in matrix $\mathbb{Y}$. The underlying partitioning procedure is then provided by the k-shape clustering algorithm for time series sequences (Paparrizos & Gravano, 2015), and canonical correlation analysis to establish the strength of interaction among the groups of variables, which allows to define strong and weakly coupled neighbors according to heuristic thresholds. The algorithmic procedure for partitioning allows to retrieve k groups of strongly coupled non-overlapping subsystems with approximately the same number of variables.

7.6. Hierarchical clustering

The study (Chen et al., 2020) introduces a cooperative DMPC framework based on topological hierarchy decomposition, aiming to optimize communication efficiency while maintaining global system performance. The theory at the basis of the approach is interpretive structural modeling (Attri et al., 2013), which allows to hierarchically structure subsystems based on their coupling strength, ensuring that strongly coupled subsystems are grouped within the same layer, while weakly coupled ones are placed in lower layers. Moreover, it is assumed, not without loss of generality, that only the upper layer influences the lower layer in a sequential cascade. This hierarchical order prioritizes the resolution of the local MPC problems, and their coordination, in the upper-layer subsystems, propagating their optimal control inputs downward, and iterating the process over the fixed down-streamed variables in the lower layer.

The architecture proposed in Chanfreut et al. (2023) has a two-layer structure. The bottom layer consists of local MPC agents controlling coalitions of loops, while the top layer dynamically clusters subsystems in the network. For this, the k-means clustering algorithm groups subsystems with similar dynamics, determined by operating parameters,

and recursive least squares estimation (Shaferman et al., 2021) adapts system parameters in real-time. Moreover, the top layer accounts for exogenous conditions to assign MPC constraints to local agents. The method allows scalability of the MPC architecture, but is sensitive to parameter estimation errors and relies on fixed cluster numbers

7.7. Input-coupled systems

An algorithmic partitioning approach for input-coupled systems is proposed in Wei et al. (2020), where the objective is to derive a novel iterative DMPC strategy with a dynamic communication topology The network is assumed to be composed by a number n of coupled linear dynamics of the form

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{j\in\mathcal{N}_i}\left[A_{ij}x_j(k) + B_{ij}u_j(k)\right]. \tag{61}$$

By using the Kalman canonical form, the state coupling can be avoided with an appropriate selection of the new subsystem states (Stewart et al., 2010), providing new input-coupled local dynamics[7] $\bar{x}_i(k+1) = \bar{A}_{ii}\bar{x}_i(k) + \bar{B}_{ii}\bar{u}_i(k) + \sum_{j\in\mathcal{N}_i}\bar{B}_{ij}\bar{u}_j(k)$. A sensitivity analysis is performed to establish the effect of the coupling variables on the optimization problem. On this basis, a threshold triggering communication between local controllers is derived. Accordingly, an algorithmic procedure determines the entries of a communication matrix at each time step, thus obtaining an event-triggered topology change for the communication networks defining the local controllers.

7.8. Hierarchical clustering for input-coupled systems

Hierarchical clustering for input-coupled systems is proposed in Wang and Koeln (2023), where a distance function induced over minimal robust positively invariant sets is used as an underlying metric for the clustering algorithm. Specifically, the hierarchical clustering of Xu and Wunsch (2005) is used to design a robust Dec-MPC, as the one of Trodden and Maestre (2017). The approach is iterative and defined for a given number of hierarchy levels, starting from the network considering each agent as an individual cluster. A tuning parameter $\alpha > 0$ is defined to perform the clustering. At each step, the minimum distance $d^{\min} = \min_{ij} d_{ij}$ is computed. Then, the procedure aggregates together the agents for which $d_{ij} > (1+\alpha)d^{\min}$. Then, the procedure is iterated for the next hierarchy level until one single agent representing the entire network is obtained.

7.9. Computational complexity and controllability

An algorithmic partitioning approach oriented at the minimization of the computational complexity of the resulting DMPC architecture while ensuring the controllability of the resulting subsystems is developed in Arastou et al. (2025). To this aim, the authors develop an algorithm for the reduction of the number of iterations $\bar{r}$ required to retrieve an (approximate) solution of a distributed optimization problem with a desired accuracy ϵ. The idea behind this approach is that by finding the partitioning that minimizes the number of iterations of the DMPC, the amount of information shared among the agents will also be minimized. In Arastou et al. (2025), the desired partitioning is obtained through the minimization of the cost function F dependent by the selected partitioning $\mathcal{P}^j$ is defined as:

$$F(\mathcal{P}^j) = \left(\log_{\beta(\mathcal{P}^j)}\frac{\epsilon}{J(\boldsymbol{x}_{(0|k)}, \boldsymbol{u}^0_{(0|k)})} - 1\right)\sum_{i=1}^{N_{P_a}} g(n_i, m_i, N, n_c), \tag{62}$$

where $f(\mathcal{C}_i) = g(n_i, m_i, N, n_c)$ is a function of the number of states and inputs of the collection $\mathcal{C}_k$, the prediction horizon N, and of the number of constraints n_c; and $J(\boldsymbol{x}_{(0|k)}, \boldsymbol{u}^0_{(0|k)})$ is the cost function for the first prediction step, evaluated with the first iteration of the control action. The minimization of (62) is sought using the Kernighan–Lin algorithm (Kernighan & Lin, 1970) based on iterative node exchange.

[7] Note that this state transformation can be already considered a partitioning of the state of the network.

8. Community-detection-based partitioning

8.1. Fundamentals and modularity metric

Community detection is a fundamental branch of modern network theory, and its scope is the identification of groups of elements in the network that have a higher probability of being strictly connected to each other w.r.t. other member in the network (Fortunato & Hric, 2016). Among the methodologies for community detection, we find optimization-based, algorithmic, dynamics-based, and consensus-based approaches, as well as methods based on statistical inference, and spectral or hierarchical clustering: an extended discussion about these topics can be found in Fortunato (2010) and Fortunato and Hric (2016). Partitioning approaches based on the quality function called *modularity* belong to the broader class of methods for community-detection in graphs (Fortunato, 2010), i.e. they are clustering methodologies, often algorithmic. In this context, modularity is a metric that has been consistently used to quantify the quality of the resulting clusters, not only in network theory, but also for control systems. Several studies in the field of partitioning for predictive control use modularity as a fundamental metric. Therefore, we treat this topic separately from other partitioning approaches.

In control theory, modularity has been applied to compute the partitioning of the graph associated with a dynamical system. The method to derive this graph has been discussed in Section 3.2. However, modularity-based partitioning can also be deployed over agent-based representations of the form Section 3.3, which is a conceptually different use case. In general, for a network with a given adjacency matrix $\mathcal{A}$, and a partition into N communities $\mathcal{P} = \{\mathcal{C}_1, \dots, \mathcal{C}_N\}$, the modularity Q index is constructed as:

$$Q = \frac{1}{2m}\sum_{ij}\left(\mathcal{A}_{(i,j)} - \frac{k_i^{\text{in}}k_j^{\text{out}}}{2m}\right)\delta_{ij}, \tag{63}$$

where $\mathcal{A}_{(i,j)}$ is the ijth element of the adjacency matrix, k_i^{in} and k_i^{out} are respectively the in- and out-degree of node i in the network, m is the total number of edges, and the binary variable δ_{ij} is equal to 1 if nodes i and j are in the same community, and zero otherwise. Modularity-based partitioning approaches all focus on finding the partitioning $\mathcal{P}$ that maximizes the modularity Q (usually, for a given number N of communities). In the remainder of this section, we will discuss how modularity-based partitioning has been used in predictive control, and provide different examples.

8.2. Maximization of modularity by iterative bipartition of the network

The most used methodology for modularity maximization in control has been presented in Jogwar and Daoutidis (2017). The approach is based on the construction of the modularity matrix $\mathcal{B}$, whose entries are defined as:

$$\mathcal{B}_{(i,j)} = \mathcal{A}_{(i,j)} - \frac{k_i^{\text{in}}k_j^{\text{out}}}{m}. \tag{64}$$

Then, the partitioning approach iteratively splits the network into two communities. To this aim, a vector $\boldsymbol{s}$ with a size equal to the number of nodes in the network is defined as follows. When a split is performed, the network $\mathcal{N}$ is divided into two communities: $\mathcal{C}_a$ and $\mathcal{C}_b$. Accordingly, the ith entry of $\boldsymbol{s}$ is defined to be equal to 1 if $i \in \mathcal{C}_a$, and -1 if $i \in \mathcal{C}_b$. The modularity associated with this new partition of the network is then:

$$Q = \frac{1}{4m}\boldsymbol{s}^{\intercal}\left(\mathcal{B} + \mathcal{B}^{\intercal}\right)\boldsymbol{s}. \tag{65}$$

The specific partitioning algorithm used to perform the modularity maximization of the basis of the iterative division is Leicht and Newman (2008), which successively divides the network into communities using approximate spectral optimization for the divisions. Fine-tuning by node shifting (Newman, 2006) is performed at each step to improve the partitioning quality.

The paper (Pourkargar et al., 2017) investigates the impact of system decomposition on the performance and computational efficiency of DMPC applied to nonlinear process networks (Liu, Chen et al., 2010). The study compares the partitioning of a network obtained through community detection with intuitive partitioning given by expert subsystem selection according to energy or technical considerations. The metrics used for comparison are the closed-loop control performance and computational burden.

Extension of the partitioning methodology (Jogwar & Daoutidis, 2017) to weighted graphs using the module of the partial derivatives of the dynamics around the operating points for nonlinear systems is proposed in Jogwar (2019). The partitioning procedure relies on a modified version of the multiway spectral community detection algorithm (Zhang & Newman, 2015) developed for unweighted graphs.

8.3. For optimization problem decomposition

An algorithmic partitioning approach for the optimization problem decomposition using community detection has been proposed in Tang, Allman et al. (2018). The optimization problem related to DMPC considered in this work is assumed to be in a "separable" form:

$$\min_{v} \; f_1(v_1) + \cdots + f_n(v_n) \tag{66}$$

$$\text{s.t. } c_j(v_1, \ldots, v_n) = 0, \;\; j = 1, \ldots, m \tag{67}$$

$$v_i \in \mathcal{V}_i, \;\; i = 1, \ldots, n, \tag{68}$$

where the scalar variables in v belong to decoupled intervals, the objective function is separable, and the coupling in the problem only arises in the equality constraints. To decompose the problem, the authors of Tang, Allman et al. (2018) use two different graph representations. In the first, they use a bipartite graph, where variables are linked to constraints according to the existence of their partial derivatives, thus capturing their functional interaction. In the second graph, they use a unipartite representation using the variables as nodes, and the number of coupling constraints as arcs. From these two graphs, it is possible to obtain adjacency matrices, and accordingly find the partitioning of these graphs that minimizes the modularity, both for unipartite (Newman & Girvan, 2004), and bipartite (Barber, 2007) representations. Modularity optimization is achieved using the Louvain fast unfold algorithm (Blondel et al., 2008).

Optimization problem decomposition based on modularity optimization is proposed in Segovia et al. (2021) through the use of optimality condition decomposition (OCD) (Conejo, 2006), to overcome the assumption that the cost function of the optimization problem must be separable to decompose it. For a given non-completely-coupled optimization problem:

$$\min_{\boldsymbol{z}} \; f(\boldsymbol{z}) \tag{69}$$

$$\text{s.t. } b(\boldsymbol{z}) \leq 0, \tag{70}$$

the OCD allows the problem to be decomposed into N subproblems, for which a relaxed formulation (Bertsekas, 1996) takes the form

$$\begin{aligned} \min_{\{z^{(i)}\}_{i=1}^{N}} \; & \sum_{i=1}^{N} f^{(i)}(z^{(i)}) + \lambda^{(i)} h^{(i)}\left(z^{(1)}, \ldots, z^{(N)}\right) \\ \text{s.t. } & h^{(i)}\left(z^{(1)}, \ldots, z^{(N)}\right) \leq 0 \qquad i \in \{1, \ldots, N\} \\ & g^{(i)}\left(z^{(i)}\right) \leq 0 \qquad i \in \{1, \ldots, N\}, \end{aligned} \tag{71}$$

where $z^{(i)}$ is the variable of the ith subproblem, $\boldsymbol{h}$ is a set of complicating constraints without which the subproblems would be independent, $\boldsymbol{g}$ are the constraints resulting from the conversion of $b(\boldsymbol{z}) \leq 0$, and λ are the Lagrange multipliers. To the problem (71) is associated the matrix of first-order Karush-Kuhn–Tucker condition (Boyd & Vandenberghe, 2004) that can naturally be interpreted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, for which modularity-based community detection can be applied. Modularity maximization is achieved through the fast unfold algorithm (Blondel et al., 2008), thus providing a decomposition of the optimization problem and consequently a partitioning of the system.

8.4. Frequency-based graph weighting

The use of a frequency-based index to perform partitioning through community detection is explored in Wang et al. (2023), where the network is represented through an input–output bipartite graph, as in Section 3.4. The edges connecting I/O variables are weighted through the linearized frequency response between each pair of variables. Specifically, the integral of the magnitude of the transfer function between two variables (u_i, y_j) for a given range of frequencies $[\omega_1, \omega_2]$ is computed as:

$$\beta_{ij} = \int_{\omega_1}^{\omega_2} \frac{|G_{ij}(j\omega)|}{\sqrt{1 + |G_{ij}(j\omega)|^2}} d(\omega), \tag{72}$$

and then a normalization is used to obtain the weights $w_{ij} = 1 - e^{-\beta_{ij}}$. This allows to retrieve a monotonically increasing weighting in the range $[0, 1]$ for all the edges. The computation of the partitioning based on this weighting is performed through a modified version of Barber's algorithm (Barber, 2007).

8.5. Time-varying graph representations

Exploration of a partitioning algorithm for time-varying systems is proposed in Arastou et al. (2025) where nonlinear dynamics of the following form are considered:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g(x(t), u(t)) \\ y(t) &= h(x(t)). \end{aligned} \tag{73}$$

For this class of systems, an associated graph representation is constructed using as weighting for the edges the partial derivatives of the dynamics w.r.t. the variables. Specifically, denoting with an arrow an edge between variables, the corresponding weights are defined as in Kravaris and Kantor (1990):

$$u_i \to x_j : \left|\frac{\partial g_j}{u_i}\right|; \quad x_i \to x_j : \left|\frac{\partial f_j}{x_i}\right|; \quad x_i \to y_j : \left|\frac{\partial h_j}{x_i}\right|. \tag{74}$$

Once all the weights are defined, the corresponding adjacency matrix A^{adj} is constructed, and accordingly, the modularity metric Q can be used for graph partitioning. The algorithm used in this case is the spectral community detection detailed in Zhang and Newman (2015).

8.6. Hierarchical approach for time-varying graphs

A hierarchical algorithmic approach for time-varying topologies is presented in Riccardi et al. (2025b, 2025c). Starting from the graph representation in Section 6.7, the partitioning problem is divided into two parts: first, a selection of fundamental and indivisible systems dynamics, called FSUs, is performed algorithmically; then the FSUs are aggregated into collections, called composite system units (CSUs), for which a controller is designed. The algorithm for this procedure is available in Riccardi et al. (2025c). Application of the algorithmic selection of FSUs allows to obtain a network structure $\mathcal{N} = \{\mathcal{S}_1, \ldots, \mathcal{S}_{N_{\text{FSU}}}\}$ from any given dynamics of the form (3). The second part of the partitioning strategy is an aggregative procedure for merging FSUs into CSUs. To this aim, a modularity-inspired metric is designed to capture

the strength of the interaction intra- and inter-CSUs, while balancing their size. These individual components of the metric are:

$$W_{C_i}^{\text{intra}} = \sum_{s,t\in\mathcal{V}_i} |w_i(s,t)| \tag{75}$$

$$W_{C_i}^{\text{inter}} = \sum_{s\in\mathcal{F}_{C_i}} \sum_{j\in\mathcal{N}_{C_i}} \sum_{t\in\mathcal{N}_s\cap\mathcal{V}_j} |w_i(s,t)| + |w_j(t,s)| \tag{76}$$

$$W_{C_i}^{\text{size}} = |C_i|^2, \tag{77}$$

where $\mathcal{V}_i$ is the set of the nodes in the set C_i, and $\mathcal{F}_i$ its frontier. Using these terms, the global metric for partitioning, named partition index, is defined as:

$$p^{\text{idx}}(\mathcal{P}) = \frac{\sum_{i=1}^{m} W_{S_i}^{\text{Intra}}}{1+\sum_{i=1}^{m} W_{S_i}^{\text{Inter}}} + \frac{\alpha}{1+\sum_{i=1}^{m} W_{S_i}^{\text{size}}}, \tag{78}$$

where α is the parameter affecting the granularity, thus allowing balancing the effect of the size of the collections in the partitioning. A greedy algorithmic procedure is used to iteratively assign the subsystems S_i to the collections C_i such that at each assignment the variation $\Delta p^{\text{idx}} = p^{\text{idx}}(\mathcal{P}^{\text{new}}) - p^{\text{idx}}(\mathcal{P}^{\text{old}})$ is maximized. The partition index defined in (78) can also be used in global search optimization (genetic algorithm), as similarly proposed in Riccardi et al. (2024b, 2024c).

9. Partitioning based on game-theoretical coalition formation

Coalitional predictive control is among the most recent formulations of non-centralized predictive control (Maestre et al., 2014). It consists of a combination of optimization-based control and game theory in which dynamical groups of agents cooperate to achieve a coordinated action to optimize some given performance criteria. At the basis of this strategy, there is the concept of *coalition formation*, explained in detail in Ray (2007), according to which agents in a network group themselves into coalitions to improve their collective outcome. In coalitional control this concept is used to obtain a distributed control strategy.

In this section, the main partitioning strategy used in coalitional predictive control will be introduced first, and then details about fundamental alternatives will be given. After, the theoretical properties of coalitional predictive control and their relation to partitioning are discussed. Various extensions and applications are presented in the remainder of the section.

9.1. The concept of coalitional control: predictive control and game theory

Consider a network $\mathcal{N}$ constituted by $N_{\mathcal{A}}$ agents, i.e. a collection $\mathcal{N} = \{\mathcal{A}_1, \dots, \mathcal{A}_{N_{\mathcal{A}}}\}$. A *coalition* C is any subset $C \subseteq \mathcal{N}$ where agents in C cooperate. To each coalition it is assigned a *characteristic function* $v(C)$, mapping the coalitions into real numbers, i.e. $v : 2^{N_{\mathcal{A}}} \to \mathbb{R}$, $v(C) \geq 0$. A *coalitional structure* $\mathcal{P}$ is a collection of disjoint coalitions covering the entire network, in other words a non-overlapping partitioning of the network $\mathcal{P} = \{C_1, \dots, C_{N_C}\}$. The value of the coalitional structure is the sum of the individual contributions of each coalition:

$$V(\mathcal{P}) = \sum_{C\in\mathcal{P}} v(C). \tag{79}$$

The objective of the *characteristic function game* (CFG) (Sandholm et al., 1999) played by the agents, and that is considered in coalitional control, is to find the coalitional structure that maximizes the total welfare:

$$\mathcal{P}^* = \arg\max_{\mathcal{P}\in\mathcal{M}} V(\mathcal{P}), \tag{80}$$

where $\mathcal{M}$ is the set of all possible disjoint partitions of $\mathcal{N}$. Various methodologies can be deployed to solve this problem, as it will be presented in the remainder of the section.

The framework of the CFG is well suited for developing distributed predictive control strategies since it is, at its core, a distributed optimization approach. One of the first works that formalizes the coalitional predictive control strategy is (Fele et al., 2017), where a large-scale system is assumed to be composed of subsystems of the form:

$$\begin{aligned} x_i(k+1) &= f(x_i(k), u_i(k)) + w_i(k) \\ w_i(k) &= \sum_{j\in\mathcal{N}_i} h(x_j(k), u_j(k)). \end{aligned} \tag{81}$$

Each of these subsystems is an agent $\mathcal{A}_i$, and it can participate in a coalition C_ℓ, such that $\bigcup_{\ell=1}^{N_C} C_\ell = \mathcal{N}$, $C_\ell \cap C_m = \emptyset\ \forall \ell, m \in \{1, \dots, N_C\}$, with N_C the number of coalitions. Each subsystem is associated with a local optimization problem:

$$\begin{aligned} \min_{\tilde{x}_{i,k}, \tilde{u}_{i,k}} \ J_i = &\sum_{j=1}^{N-1} J_{\text{s}}(x_i(j|k), u_i(j-1|k)) \\ &+ J_{\text{f}}(x_i(N|k), u_i(N-1|k)) \\ \text{s.t.} \quad & x_i(k+1) = f(x_i(k), u_i(k)) + \hat{w}_i(k) \\ & x_i(0|k) = x_i(k) \\ & g_i(\tilde{x}_{i,k}, \tilde{u}_{i,k}) \leq 0, \end{aligned} \tag{82}$$

where $\hat{w}_i$ is an estimate of the dynamical coupling of x_i with its neighboring subsystems, and $\tilde{x}_k$, $\tilde{u}_k$ are the state and input sequences defined over the prediction horizon N for a time step k. A coalition C_ℓ is formed only if the value of the cost associated with the coalition, i.e. J_ℓ, is lower than the sum of the costs of the individual subsystems. Thus, the coalition formation condition is:

$$J_\ell^* < \sum_{i\in C_\ell} J_i^*. \tag{83}$$

In the framework of CFG, the simplest characteristic function associated with a coalition C_ℓ is $v(C_\ell) = J_\ell^*$. In this case the coalition formation problem consists in finding the optimal coalitional structure $\mathcal{P}^* = \arg\max \sum_{C_\ell\in\mathcal{P}} v(C_\ell) = \sum_{\ell=1}^{N_C} J_\ell^*$, with a number N_C of coalitions. This problem is known to be NP-Complete (Sandholm et al., 1999), inheriting the same complexity of the general partitioning problem.

The underlying principle of coalition formation described above is shared among all coalitional control strategies, and variations are present in the definition of the characteristic function, the individual payoffs, the implementation of the local MPC controllers, the computation of ordering maps sorting agents costs, and the aggregation algorithm. In the remainder of the section, we report variations, extensions, and applications of the partitioning approach found in coalitional control literature.

9.2. Foundational works

Coalitional predictive control is effectively formalized in the seminal work (Fele et al., 2017). To overcome the computation complexity associated with the general coalition formation approach described in the previous section, the partitioning problem is addressed by looking at the coalitional structure $\mathcal{P}$ where the participation preference of each agent $\mathcal{A}$ is sorted according to their Pareto ordering. This is achieved by first using the Shapley value (Shapley, 1953) to compute the individual payoffs of each agent $\mathcal{A}$ in each possible subset of agents $S \subseteq \mathcal{N}$, that for agent $\mathcal{A}_i \in S$ is defined as:

$$\phi_{\mathcal{A}_i}^S = \sum_{C\subseteq S\setminus\mathcal{A}_i} \frac{|C|!(|S|-|C|-1)!}{|S|!} \left[v(C\cup\mathcal{A}_i) - v(C)\right]. \tag{84}$$

Using the Shapley value it is possible to build a mapping $\Phi : \mathcal{N}\times 2^{\mathcal{N}}\times\mathbb{Z}\to\mathbb{R}$ for each agent in each possible coalition, i.e. at each time step k a function $\Phi(\mathcal{A}_i, C_j, k)$ is available. The function Φ provides for each agent their preferred participation order into coalitions. Accordingly,

agents can autonomously organize into the coalitional structure[8] $\mathcal{P}^{\Phi}$. The dynamic coalition formation is guided by an individual payoff Φ coinciding with the energy exchange with the main grid.

A first extension of Fele et al. (2017) is found in Fele et al. (2018), which proposes a coalitional predictive control strategy with self-organizing agents. The coalition formation strategy is based on a negotiation protocol allowing agents to autonomously form coalitions based on expected performance improvements and cooperation costs. In particular, the coalition formation problem is framed as a transferable utility game (Cesco, 1998; Sandholm et al., 1999; Stearns, 1968), where agents decide to merge or separate dynamically using a bargaining protocol. The coalitional benefit is considered under the assumption of individual rationality, described in the following. Consider two coalitions C_1, C_2, and the value of their individual and aggregated characteristic functions, i.e. $v(C_1)$, $v(C_2)$, and $v(C_1 \cup C_2)$. Also, consider the value associated with each of the players in the coalition, denoted by $v(C_1 \cup C_2)|_{(i)}$ for $i = 1, 2$, and defined such that $v(C_1 \cup C_2)|_{(1)} + v(C_1 \cup C_2)|_{(2)} = v(C_1 \cup C_2)$. Then the merger of $v(C_1)$, $v(C_2)$ occurs if and only if the condition $v(C_1 \cup C_2)|_{(i)} \leq v(C_i)$ holds for both $i = 1, 2$, which is known as individual rationality. The value associated with a player $v(C)$ is then considered as an economic index, a utility that can be transferred. Consequently, a bargaining procedure is designed to merge the coalitions considering that, when aggregating two coalitions, the value $v(C_1) + v(C_2) - v(C_1 \cup C_2)$ is a surplus that can be reallocated between the remaining agents.

Another bottom-up aggregative procedure for coalitions has been devised in Maestre and Ishii (2017) where a PageRank (Brin & Page, 1998; Ishii & Tempo, 2014) approach is used as the metric to guide local node exchanges among coalitions. For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the PageRank associated with each node $i \in \mathcal{V}$ is a scalar $p_i \in [0, 1]$, s.t. $\sum_{i \in \mathcal{V}} p_i = 1$. Given the neighborhood $\mathcal{N}_i$ of node i, its PageRank value is computed as $p_i = \sum_{j \in \mathcal{N}_i} p_j / n_j$, where p_j is the value associated with node j, and n_j its number of edges. Once the values p are known for all the nodes, a weighting of the links is performed assigning to each ϵ_{ij} a weight $w_{ij} = p_i / n_i$. The distributed computation of the PageRank is performed using the algorithm (Ishii & Tempo, 2010), then an algorithm to aggregate nodes into coalitions using iterative aid requests.

A combination of the methodologies (Fele et al., 2017) and (Maestre & Ishii, 2017) is found in Muros et al. (2018) where a randomized method for the estimation of the Shapley value is applied. Specifically, the Shapley value defined as the vector $\phi(\mathcal{N}, v)\ \forall i \in \mathcal{N}$, for the game induced over the set of agents $\mathcal{N}$ and for a characteristic function v (coinciding with the stage cost of the local MPC), is used to introduce a weighting of the links among agents, which is defined for the undirected link $ij \in \mathcal{E}$ as $w_{ij} = \phi_i(\mathcal{N}, v)/|\mathcal{E}_i| + \phi_j(\mathcal{N}, v)/|\mathcal{E}_j|$. To address the problem of the combinatorial explosion associated with the computation of the Shapley value associated with all possible coalitions, randomized methods (Castro et al., 2009; Ishii & Tempo, 2010) are proposed to estimate it. In particular, using the modified definition of the Shapley value given in Castro et al. (2009), an estimation of its value is given for a set of q samples of all possible coalitions, giving an approximation of the value, whose efficient estimate is distributed as $\tilde{\phi}_i(\mathcal{N}, v) \sim N(\phi_i, \sigma^2_{\phi_i}/q)$, with bounded error.

9.3. Technical extensions: feasibility, stability, robustness

Theorems for the stability and recursive feasibility (Mayne et al., 2000) of a coalitional predictive control formulation have been proposed in Baldivieso Monasterios and Trodden (2021). The aggregation of coalitions is achieved through a consensus procedure, where for each coalition C_i in a given state x a consensus optimization problem is defined as:

$$\min_{C_i \in \mathcal{M}} J_i(C_i, C_{-i}, x) = J_i^{\text{consensus}}(C_i, C_{-i}, x) + \rho J_i^{\text{power}}(C_i, x), \tag{85}$$

and $C_{-i} \triangleq \{C_j\}_{j \in \mathcal{N}_i}$ is the set of possible neighboring coalitions. In this optimization problem, the term $J_i^{\text{consensus}} = 0$ if coalitions C_i and its neighbors agree on the current arrangement into coalitions, and the term J_i^{power} weighted by the scalar ρ represents the effect of coalition C_i on neighbors opinions (Muros et al., 2017). The consensus optimization is achieved through an algorithm that leverages the theory of finite exact potential games (Monderer & Shapley, 1996).

Another extension is found in Chanfreut, Maestre, Ferramosca et al. (2022), where tracking of target sets is achieved. Coalitions are formed to enlarge the domain of attraction of MPC, but when sufficient, the decentralized formulation is used. The underlying partitioning strategy is hierarchical, where partitioning is executed at a slower time scale over a heuristic selection of possible communication topologies. The coalitional scheme is defined by Maestre et al. (2014). In particular, given the set $\mathcal{M}$ of all possible communication topologies, and for a partitioning $\mathcal{P} \in \mathcal{M}$, the characteristic is defined as:

$$V(\mathcal{P}, x_{\mathcal{P}}) = (x_{\mathcal{P}} - x_{\Gamma})^{\intercal} P_{\mathcal{P}} (x_{\mathcal{P}} - x_{\Gamma}) + c|\mathcal{E}_{\mathcal{P}}|, \tag{86}$$

where $x_{\mathcal{P}}$ is the aggregated state of all the coalitions at time step k, x_{Γ} is the Chebyshev center of the target set, $|\mathcal{E}_{\mathcal{P}}|$ is the number of communication links enabled in the partitioning $\mathcal{P}$, $c > 0$ is a scalar, and $P_{\mathcal{P}}$ is a positive definite matrix.

9.4. Market-based partitioning

A market-based coalition formation approach applied to coalitional predictive control is introduced in Masero et al. (2022). The strategy is inspired by other market-based approaches (Son et al., 2004), and results in a hierarchical coalitional control strategy. For a given objective function J of the plant, a quadratic sum of the output and of the control variable q, the market-based coalitional strategy is implemented by defining for each agent $i \in \mathcal{N}$ in the plant, a utility value $U_i(\cdot) = -J_i(\cdot)$ that the agent i can supply or demand to purchase or sell a quantum of input Δq. Accordingly, the set of agents is split into two disjoint subsets $\mathcal{L}_{\text{s}}$, $\mathcal{L}_{\text{d}}$ of supply and demand agents, with respective utilities. Then, the utility is computed and classified according to the two groups for each agent or coalition. This way, the requests can be sorted in descending and ascending order for demand and supply, and trades are performed according to this matching. The hierarchical coalition formation procedure is then implemented starting from the coalition formed by individual agents, and runs periodically according to a fixed time step bigger than the control step. Heuristics ensure the terminability of the algorithm.

Feedforward Neural Networks (NNs) (Fine, 2006) are used in Masero, Ruiz-Moreno et al. (2023) to reduce the computational complexity of the market-based hierarchical formulation introduced in Masero et al. (2022). Specifically, in Masero, Ruiz-Moreno et al. (2023) sets of NNs are used with two different scopes in cascade. The first set of NNs uses information about states and disturbances to approximate the values of the utilities of supply and demand agents. These are used to implement the market-based coalition formation. Then, a second set of NNs, using the same information and considering the coalition obtained, approximate the value of the input q for the coalitions, that can group at most three loops. The drawbacks of this strategy arise from the defining technical characteristics of NNs, which include the necessity of rich enough data to perform the training, the inability to provide suitable outputs when the operating conditions of the plant are distant from the training set, and the lack of guarantees for constraint satisfaction.

[8] The partitioning $\mathcal{P}^{\Phi}$ does not necessarily coincide with the optimal partitioning $\mathcal{P}^*$ in terms of global minimization of the value of the cost function $J(\tilde{x}_k, \tilde{u}_k, \tilde{\delta}_k)$ in (46).

9.5. Further extensions

Predicted topology transition is proposed in Masero, Maestre et al. (2021) as an evolution of the work in Fele et al. (2017). The method extends Coal-MPC by incorporating a transition horizon variable, which optimizes the timing of topology changes over the prediction horizon. Unlike previous coalitional control methods that switch coalition structures instantaneously, this approach gradually transitions between topologies, allowing agents to anticipate and optimize their control actions accordingly. The strategy also belongs to hierarchical coalitional control, where the upper layer, working at a lower rate, is designed to obtain the desired coalition and the transition horizon.

Pairwise clustering is proposed in Masero, Frejo et al. (2021) where agents are grouped in couples, yielding to a hierarchical control approach. In the upper layer, at each time step k, the measurement of the inputs in each agent at the previous time step is collected into a vector $q_{k-1}^{\text{measured}}$. This vector is then sorted in ascending order, giving q_{k-1}^{sorted}. Then, the partitioning of the plant is obtained by coupling together the first and last elements of q_{k-1}^{sorted} and removing them from the vector until no further assignments are possible.

The problem of resource sharing under partitioning is addressed in Sánchez-Amores, Martinez-Piazuelo et al. (2023), where a prior partition $\mathcal{P} = \{\mathcal{C}_1, \dots, \mathcal{C}_{N_p}\}$ of the system is assumed to be given, e.g. using one of the techniques in Masero, Frejo et al. (2021), Masero et al. (2022) and Masero, Ruiz-Moreno et al. (2023). The problem of distributing the shared resource is solved using a population-dynamics-assisted resource allocation strategy (Barreiro-Gomez & Tembine, 2018; Martinez-Piazuelo et al., 2022), specifically a Smith population dynamics with carrying capacities (Barreiro-Gomez et al., 2018). Following the hierarchical coalitional control methodology introduced in Masero et al. (2022), the resource allocation (for a fixed partitioning) is performed at a slower time scale.

9.6. Partitioning for input-coupled systems

The use of coalitional predictive control for systems with coupled input dynamics is found in Masero et al. (2020b), where the following input-coupled agent representation is considered:

$$x_i(k+1) = A_i x_i(k) + \sum_{j \in \mathcal{C}_i} B_{ij} u_{ij}(k) + \omega_i(k), \tag{87}$$

with $u_{ij} = -u_{ji}$, and ω_i is a disturbance. The underlying coalitional formation approach is a hierarchical methodology of the form (Fele et al., 2017), where in the upper layer a new coalitional structure is assigned according to a fixed time step longer than the control sampling time. In this case, the computational complexity of evaluating the best topology is reduced by considering as candidate successors only the allocations $\mathcal{P}^{\text{next}}$ that have a Hamming distance of one from the current configuration $\mathcal{P}^{\text{current}}$, i.e. they differ from only one link allocation.

A further extension of coalitional predictive control for coupled input dynamics has been proposed in Sánchez-Amores et al. (2022). In this work, couplings in the inputs among agents are decomposed into private and public variables, a feature detailed in La Bella et al. (2019). This approach is used because it allows more flexibility in the computation of the control action w.r.t. robust approaches as tube-based MPC that is more conservative.

An extension of Sánchez-Amores et al. (2022) is found in Sánchez-Amores, Chanfreut et al. (2023), where a robust tube-based formulation of the controller is proposed (Mayne et al., 2005). Additionally, in Sánchez-Amores, Chanfreut et al. (2023) the presence of communication links is event-driven, i.e. communication links are activated only if scaling factors exceed predefined thresholds that allow to establish a trade-off between performances and communication burden.

A further advancement in coalitional control for input-coupled dynamics is achieved in Masero, Baldivieso-Monasterios et al. (2023), where a robust strategy allowing plug-and-play capabilities is devised. The approach is based on an evolution of public and private factors introduced in Trodden and Maestre (2017) and already employed in Sánchez-Amores et al. (2022).

10. Heuristic partitioning

In this section, we present partitioning techniques that have been developed for specific applications. These techniques, while not broadly applicable at the current stage, might eventually be generalized. In addition, they still find relevance to their field, for which are readily available.

Partitioning for Dec-MPC of wide-area power systems is investigated in Jain et al. (2018). The technique is heuristic and based on the use of the modal participation matrix that highlights the effects of each generation on each dominant mode in low-frequency oscillations. The partitioning technique allows overlapping partitioning, giving both the nature of the dynamical couplings and the use of a DMPC strategy (Alessio & Bemporad, 2007). The approach is applied to the Northeast Power Coordinating Council nonlinear power system model (Rogers, 2000), comprehending 48 electrical machines and 140 buses, showing the performance and the resilience of the network for two different partitionings compared to centralized control.

Partitioning for wind farms is proposed in Ye et al. (2019), where a HMPC strategy is proposed. The partitioning strategy is performed on the highest level of the hierarchy every 15 min. Based on a forecast of the wind characteristics for the next 20 min, an optimization strategy is deployed to cluster the wind turbines in one of 12 categories based on the possible load operating conditions the turbines can experience. The proposed HMPC strategy was validated over a modified version of the IEEE One Area RTS-96 network (Grigg et al., 1999), and compared with conventional dispatch and schedule allocation algorithms, achieving significantly better performance.

A strategy for partitioning vehicle platoons is implemented in Liu et al. (2019), with the objective of deploying a noniterative two-level DMPC architecture ensuring closed-loop stability for an optimization problem with coupled cost functions and constraints. The partitioning strategy is based on the assumption that the cooperation set of vehicles $\mathcal{V}$ can be divided into groups that belong to two main conceptual categories, i.e. dominant and connecting clusters. The algorithmic partitioning allows vehicles to perform the operations of joining and leaving a platoon on the basis of this group classification. The DMPC strategy is then designed around this partitioning approach, ensuring stability and feasibility. Validation of the approach is performed for a platoon of four vehicles, and compared against CMPC, showing minimal loss in performance.

A strategy for event-triggered partitioning of microgrids is developed in Ananduta and Ocampo-Martinez (2021), where the economic dispatch problem for energy production is addressed. The power network is considered to be constituted of microgrids that are considered self-sufficient systems, i.e. they do not exchange energy with their neighbors in nominal operating conditions. However, if this generative autonomy is lost, re-partitioning of the network is triggered, leading to a new definition of the microgrids. This re-partitioning is performed through a communication protocol, which evaluates the best node exchange among the microgrids that minimizes the individual outcomes in economic terms, while ensuring self-sufficiency. The approach is validated on the PG&E 69-bus distribution network. The simulation results show that during peak hours all microgrids should join into a single agent to satisfy the demand, whereas during off-peak hours they can split into multiple coalitions.

The paper (Huanca et al., 2023) proposes a distributed Switching Model Predictive Control (SMPC) strategy for quadrotor UAV swarm aggregation incorporating collision avoidance. Teams of UAVs are selected using a clustering strategy, and local controllers solve the SMPC problem sequentially (Christofides et al., 2013). The clustering approach is based on the sphere packing problem (Conway & Sloane, 1988). A cluster of UAVs is selected according to the positions of UAVs in space (Gauci et al., 2014), assuming these are always available. In the sphere packing problem, the objective is to find the arrangement of non-overlapping spheres so that they occupy the largest possible

fraction of space. Solutions are available in the literature for this problem (Conway & Sloane, 1988). The approach is validated with a group of 150 UAVs, using both centralized and distributed control strategies for the aggregation. The proposed distributed SMPC can achieve comparable aggregation performance w.r.t. its centralized counterpart while drastically reducing computation time.

11. Applications and case studies

In this section, we develop a classification that relates the partitioning methodologies found in the literature to the systems used for their validation. In the resulting Table 8, for each application system, partitioning methodologies are classified according to Fig. 4. When possible, we also provide references to more standardized test cases for their direct use in the development of further strategies. After this classification, we briefly describe the known case studies in the literature and the works that have been developed using them, providing information about the resulting control architectures and performance when available. Finally, for community-detection-based and game-theoretic oriented methods, we discuss additional applicative works that are of interest.

11.1. Classification and benchmark systems

From Table 8, we note that many works have been developed for power systems. However, if we consider standard generation and transmission systems, no specific case study has been consistently used to derive partitioning techniques. Therefore, it is difficult to quantitatively compare different works. An exception in this sector is the parabolic-trough plant ACUREX (Gallego & Camacho, 2012; Gálvez-Carrillo et al., 2009), for which many Coal-MPC strategies have been developed.

Several other applications are reported in Table 8, all used in the development of a specific partitioning methodology for the application of non-centralized control. Especially for transportation networks, we observe a notable lack of studies in partitioning for NCen-MPC of urban traffic, freeway transportation, and railway networks (Luan et al., 2020). The other case studies are, in general, smaller systems that can be used for the development of strategies, but do not stress the scalability of the approaches.

Several other large-scale application fields can be considered for studies in partitioning, such as swarms of mobile robots or autonomous maritime vehicles (Zhou et al., 2020), automated agricultural systems, district heating (Blizard & Stockar, 2025), satellite constellations (Curzi et al., 2020), and advanced industrial processes (Galloway & Hancke, 2013). Some of the applications listed can be found, for example, in the recent work (Pedroso et al., 2025) about the design of large-scale systems, or in the set of benchmarks proposed in Maestre and Ocampo-Martinez (2025).

11.2. Analysis of the case studies

In this section, we discuss in more detail how the case studies in Table 8 have been used in the literature to validate partitioning approaches for NCen-MPC methods. The works are divided by application domain, where different partitioning classes have been used for each of them.

Power systems. Regarding energy generation and transmission networks, we find the implementation of the coalitional control approach (Fele et al., 2017) applied to energy management in smart grids, specifically to optimize local energy trade among consumer nodes with distributed generation and storage capabilities. In Fele et al. (2017), prosumers (producers-consumers) (Larsen et al., 2014) cooperate to reduce power dependence from the main grid while minimizing energy exchange costs and transmission losses among them. Simulation results illustrate how coalitional structures evolve over time, showing how coalitional trade reduces overall costs compared to grid-dependent strategies, as prosumers can access more favorable internal energy prices. Also the Coal-MPC strategy in Fele et al. (2018) is applied for wide-area control of power networks (Chakrabortty & Khargonekar, 2013), showing the ability of the architecture to adapt to topological changes that may arise with faults or network extensions. An HMPC approach with local Dec-MPC is deployed in La Bella et al. (2022) to control local power clusters independently. Local requests of energy activate a supervisory layer if clusters cannot satisfy the demand. Further features, such as using energy storage systems, multiple time scales, and ADMM distributed computations in the supervisory layer, are detailed in La Bella et al. (2022). The approach is implemented on the IEEE 118-bus, showing the online clustering capabilities of the approach.

An important case study in the energy sector is the parabolic-trough plant ACUREX, located in Plataforma Solar de Almería (Gallego & Camacho, 2012; Gálvez-Carrillo et al., 2009) composed by 10 loops, and its scaling to 100 loops, for which many Coal-MPC strategies have been developed. A parabolic-trough solar collector field is a system composed of many loops of parabolic collectors focusing heat on a trough flowed by the heat transfer fluid (HTF). This fluid is thus heated and can be used for electrical energy generation. The objective of a control strategy applied to this system is to maximize the thermal power output by regulating the flow q of the HTF across the loops, where the dynamics of the plant is nonlinear and subject to disturbances caused, e.g. by the variability in atmospheric conditions. For example, a direct pairwise matching approach in Masero, Frejo et al. (2021) to cluster loops of the plants in couples is motivated by the fact that loops with a deficit of flow rate can benefit from those with excess flow. The approach has been proven to outperform Dec-MPC, approaching CMPC performance while significantly reducing computation time. The market-based control architecture developed in Masero et al. (2022) is also validated on the model of the real-world collector field ACUREX. Comparison strategies include PI control, two different CMPC strategies, and the control strategy based on loop-pair clustering devised in Masero, Frejo et al. (2021). According to the simulation results, the market-based coalitional predictive control is the best-performing strategy with a gain of 12.51% w.r.t. PI control, outperforming also the CMPC implementation with 0.37%. Additionally, an analysis of the computational burden is performed. In practice, the CMPC strategy is not deployable because its computation time exceeds the operating time step of the plant. On the contrary, market-based coalitional control is fast enough to be potentially scaled up to a plant of about 300 loops while maintaining the same performance gains. The coalitional controller in Masero, Ruiz-Moreno et al. (2023) is compared with the nonlinear coalitional controller developed in Masero et al. (2022). The NN-based coalitional controller (Masero, Ruiz-Moreno et al., 2023) shows a performance that is comparable with the one obtained in the nonlinear implementation (Masero et al., 2022), but providing a considerable reduction in the computation time needed to compute the control action and the partitioning of the network with a reduction up to the 99% w.r.t. the time required in the NLin-MPC implementation. Also the approach in Sánchez-Amores, Martinez-Piazuelo et al. (2023) is validated on a 100-loop implementation of ACUREX and compared with CMPC. The results show a negligible loss in performance while significantly reducing the computation time required to retrieve the control action. The study (Chanfreut et al., 2023) introduces a hierarchical clustering-based MPC strategy for optimizing heat transfer fluid flow rates in solar parabolic trough plants (Boukelia & Mecibah, 2013). Simulations performed on 10-loop and 80-loop plants show significant effectiveness of the technique and minimal performance loss.

Water systems. Water network control is another field that has seen extensive application of partitioning strategies. In this case, we distinguish between water-tank systems, which are usually small-scale test cases used to validate the viability of the approaches, and large-scale water distribution networks, among which the Barcelona drinking

Table 8
Application fields of the partitioning techniques for NCen-MPC, classified by sector. When available, benchmark systems have been reported.

Sector	Specific application	Partitioning techniques
Power systems	Six-area power system	Chen et al. (2020)
	Smartgrids, 8 (check) prosumers: (Larsen et al., 2014; Paauw et al., 2009)	Fele et al. (2017)
	Wide area power network: (Chakrabortty & Khargonekar, 2013) The EEA-ENB: (Riccardi et al., 2024a, 2025a) PG&E 69-bus distribution network IEEE 118-bus IEEE 123 node test feeder	Fele et al. (2018) Riccardi et al. (2024b, 2024c) Ananduta and Ocampo-Martinez (2021) La Bella et al. (2022) Wang et al. (2022)
	Nonlinear power system: (Rogers, 2000), 48 machines, 140 buses	Jain et al. (2018)
	Parabolic-trough plant: ACUREX model, 100 loops (Gallego & Camacho, 2012; Gálvez-Carrillo et al., 2009)	Chanfreut et al. (2023), Masero, Frejo et al. (2021), Masero et al. (2022), Masero, Ruiz-Moreno et al. (2023) and Sánchez-Amores, Martinez-Piazuelo et al. (2023)
Water systems	4-tanks system: (Alvarado et al., 2011)	Wei et al. (2020) and Segovia et al. (2021)
	8-tanks system	Masero, Maestre et al. (2021), Maxim et al. (2023), Sánchez-Amores et al. (2022) and Sánchez-Amores, Chanfreut et al. (2023)
	16-tanks system: (Maestre et al., 2015)	Núñez et al. (2015) and Maestre and Ishii (2017)
	Barcelona drinking water transport network: (Ocampo-Martinez et al., 2009)	Barreiro-Gomez et al. (2019), Muros et al. (2018), Ocampo-Martinez et al. (2012, 2011) and Segovia et al. (2021)
	Shanghai water distribution network Richmond water distribution network: (van Zyl et al., 2004) Dez irrigation canal: (Isapoor et al., 2011; SOBEK, 2000)	Zhang et al. (2019) Arastou et al. (2025) Fele et al. (2014)
Chemical systems	2 CSTR series: (Bakule, 2008; Venkat, 2006)	He and Li (2023), Kamelian and Salahshoor (2015) and Tang, Pourkargar et al. (2018)
	2 CSTR series and flash tank separator: (Christofides et al., 2013; Liu, Muñoz De La Peña et al., 2010; Liu et al., 2009; Stewart et al., 2010)	Pourkargar et al. (2017), Rocha et al. (2018), Tang, Allman et al. (2018) and Wang et al. (2023)
	Tennessee Eastman problem: (Downs & Vogel, 1993; Lyman & Georgakis, 1995), five operation units	Xie et al. (2016)
	Benzene alkylation process: 4 CSTR and flash tank separator Amine gas sweetening plant Air separation process	Pourkargar et al. (2019) and Arastou et al. (2025) Moharir et al. (2018) Wang et al. (2023)
Wind farms	12-turbine wind farm 20-turbine wind farm, NREL 5-MW 25-turbine farm, 1.5 MW	Zhao et al. (2023) Lin et al. (2020) Changqing et al. (2022)
	42-turbine farm, NREL-5 MW: (Jonkman et al., 2009), SimWindFarm (Grunnet et al., 2010)	Siniscalchi-Minna et al. (2020)
	IEEE One Area RTS-96 network: (Grigg et al., 1999)	Ye et al. (2019)
Transportation systems	4-vehicles platoon: (Zhu et al., 2020)	Liu et al. (2019) and Maxim and Caruntu (2022)
	Urban transportation network: (De Oliveira & Camponogara, 2010), 8 intersections	Chanfreut, Maestre, Hatanaka et al. (2022)
	Jinan road network	Guo et al. (2019)
	15 km freeway stretch: (Messmer & Papageorgiou, 1990), METANET model	Chanfreut et al. (2021a)
Mechanical systems	Mass-spring-damper chain, 4 elements	Baldivieso Monasterios and Trodden (2021)
	(4 + 1)-trucks, connected with springs and dampers: (Trodden & Maestre, 2017)	Masero, Baldivieso-Monasterios et al. (2023)
	12-trucks, connected with springs and dampers: (Riverso & Ferrari-Trecate, 2012; Trodden & Maestre, 2017)	Chanfreut, Maestre, Ferramosca et al. (2022)
Smart buildings	8 rooms temperature regulation 20 thermal zones control: (Chandan & Alleyne, 2013)	Zheng et al. (2018) Atam and Kerrigan (2021)
Abstract networks	43 agents flow system: (Koeln & Alleyne, 2017) Random 50 systems, modular 64 systems, hybrid	Wang and Koeln (2023) Riccardi et al. (2025b, 2025c)
Railway networks	Dutch railway network: (Kersbergen, Rudan et al., 2016)	Kersbergen, van den Boom et al. (2016)
Telecommunication systems	Next generation cellular networks: (Auer et al., 2012)	Masero et al. (2020b)
Industrial plants	Walking beam reheating furnace system	Chen et al. (2020)
Process plant	Refinery: gas-to-liquid process, hydrocracking process	Tang et al. (2023)
Aerial vehicles	Group of 150 UAVs	Huanca et al. (2023)
Cyber-physical systems	4-agents chain	Maxim and Caruntu (2021)

water transport network (Ocampo-Martinez et al., 2009) is surely the most commonly used test case among different NC-MPC approaches. In Ocampo-Martinez et al. (2012), the ϵ-decomposition is applied to the Barcelona drinking water network, incorporating a heuristic selection of ϵ, and a hierarchical Dec-MPC strategy is applied to the resulting three-subsystem network. The architecture is validated against a CMPC controller implementation showing an overall performance loss always smaller than 2%, with a reduction of computation times up to 35%. Also the partitioning methodology in Muros et al. (2018) is validated over the Barcelona drinking water transport network by applying coalitional predictive control and comparing it against CMPC, showing how it can outperform Dec-MPC and other decentralized control architectures. In addition, a DMPC approach based on density-dependent population games (Sandholm, 2010) is used in combination with the partitioning approach developed in Barreiro-Gomez et al. (2019) and tested over the Barcelona drinking water transport network. Another notable work in water distribution networks is Zhang et al. (2019), where the network of Shanghai is considered, using 800 samples of its state and input variables captured every 10 min from 44 sensors in the network. Different partitionings are obtained by varying the parameters of the algorithm, but the one providing 6 groups is selected since it gives the minimum variance. Simulations are performed to compare the proposed enhancing DMPC strategy with the Dec-MPC approach defined in Ocampo-Martinez et al. (2012, 2011). Overall, the strategy proposed by Zhang et al. (2019) allows to achieve a reduction in the water pressure of the network, while ensuring stability and robustness, thus reducing leakages and energy requirements. The approach in Arastou et al. (2025) is applied to the control of a simplified version of the Richmond water distribution network, Yorkshire, UK (van Zyl et al., 2004), using a flow-based graph representation. The simulations show how the DMPC strategy applied to different network partitionings always ensures a negligible loss in performance, while showing computation times that gradually decrease with a higher number of sets in the partition.

Regarding water-tank systems, the methodology developed in Núñez et al. (2015) is validated on a 16 water tanks system (Maestre et al., 2015), showing how the optimal partitioning is affected by the change in operating conditions. The coalitional predictive control strategy (Maestre & Ishii, 2017) is also developed for the same 16 water tanks system, and compared against CMPC, Dec-MPC, and the DMPC scheme (Núñez et al., 2015). The strategy proposed in Maestre and Ishii (2017) is the best performer in terms of optimality gap w.r.t. CMPC, after parameters calibration. The DMPC strategy proposed in Wei et al. (2020), instead, is validated for the four-tank water system (Alvarado et al., 2011; Distributed model, 2015) against cooperative DMPC with static topology, effectively reducing the communication burden. Another method validated on a four-tanks system is the OCD-DMPC approach proposed in Segovia et al. (2021). However, the validation against other MPCs is only qualitative. For eight-tanks water systems, we report the strategy in Masero, Maestre et al. (2021), where the approach can reduce communication and coordination costs of coalitional schemes while maintaining performance close to CMPC. For a similar case study, the work in Sánchez-Amores et al. (2022) shows how varying the parameters of the partitioning strategy developed allows for balancing the communication burden with the performance loss. Finally, the approach in Sánchez-Amores, Chanfreut et al. (2023) is also validated using an eight-tank water system against centralized and Dec-MPC. The simulation results show that coalitional control can outperform Dec-MPC while approaching CMPC performances with a reduction of 83% in terms of communication cost.

Chemical plants. Chemical systems have been the subject of deep studies regarding partitioning, given the complexity of the associated dynamics. We report the presence of many system configurations involving CSTRs and separators, e.g. Bakule (2008), Liu et al. (2009), Stewart et al. (2010) and Venkat (2006) among others. Also, in this case, no single benchmark system has been used consistently in the literature; rather, there are many different similar configurations that complicate the process of direct comparison of partitioning strategies. The partitioning approach developed in Kamelian and Salahshoor (2015) is used in the deployment of a Dec-NLin-MPC strategy over an industrial chemical plant constituted by two continuous stirred-tank reactors in cascade (Bakule, 2008; Venkat, 2006). The results show that the decentralized approach proposed in Kamelian and Salahshoor (2015) has a performance comparable to C-NLin-MPC, and superior performance w.r.t. the Dec-NLin-MPC approach proposed in Venkat (2006). The method in Xie et al. (2016) is validated on a chemical plant with five operation units known in the literature as the Tennessee Eastman problem (Downs & Vogel, 1993; Lyman & Georgakis, 1995). Partitioning of the latter is executed on a linearized version of the plant around operating points generated through a stabilizing control action (McAvov & Ye, 1994). No control validation of the proposed DMPC architecture is performed. In Pourkargar et al. (2017) the analysis is conducted on a reactor-separator process, where sequential and iterative DMPC formulations (Christofides et al., 2013) are compared against CMPC. The approach in Tang, Allman et al. (2018) is validated for control of a reactor-separator process (Liu et al., 2009; Stewart et al., 2010), with two reactors in series and a separator. The approach deployed is an ADMM-based DMPC (Bertsekas, 1999), and is validated against nonlinear CMPC. The results show how the DMPC implementation can outperform CMPC for this nonlinear setting while reducing computation time by more than 50%. Among the contributions of the paper (Rocha et al., 2018) there is the derivation of two DMPC techniques, cooperative and non-cooperative, both working on linearized versions of the models. The viability of the approach is demonstrated for the reactor-separator process (Stewart et al., 2010), with two reactors in series and a separator. The approach in Tang, Pourkargar et al. (2018) is applied for deploying a noncooperative and iterative DMPC control scheme (Liu, Chen et al., 2010) over a reactor-separator process with two continuously stirred tank reactors in series (Pourkargar et al., 2017). Different decompositions of the networks are achieved, and results are compared against CMPC through a quality index normalizing the performance-computation-time product w.r.t. CMPC. This quality index is used to determine the best partitioning of the network. The partition resulting from the algorithm developed in Wang et al. (2023) is used to deploy DMPC over two different case studies, and compared with CMPC and DMPC with partitioning computed using the conventional modularity maximization. The first experiment involves a reactor separator process consisting of two continuously stirred tank reactors and a flash separator (Liu, Muñoz De La Peña et al., 2010); the second is an air separation process. The empirical results show how different decompositions of the network impact the performance of the DMPC, showing that also frequency-based modularity maximization is not always the best choice, which is in line with the concept that modularity maximization does not provide by itself the best partitioning in terms of performance. Additionally, the technique proposed only works with linear systems. The case study considered in Arastou et al. (2025) is the benzene alkylation process using four continuous stirred tank reactors and a flash separator controlled through the DMPC strategy developed in Pourkargar et al. (2019), which also involves the partitioning of the process using community detection. The strategy developed in Arastou et al. (2025) shows an improvement in the performance up to 26.9% w.r.t. the one in Pourkargar et al. (2019).

Wind farms. For this application, we found different studies in partitioning, with various topologies, turbine models, and operating conditions. The approach in Lin et al. (2020) is deployed on a farm with 20 NREL 5-MW wind turbines (Jonkman et al., 2009), modeled using SimWindFarm (Grunnet et al., 2010), and obtaining four clusters. The control approach is hierarchical and employs a proportional controller in the lower layer and an MPC in the upper layer, where in the latter, all the clusters are aggregated into a single performance index. Simulation

results show how the proposed strategy outperforms both conventional PD control and CMPC, while reducing computation times. The approach in Siniscalchi-Minna et al. (2020) is validated on a wind farm with 42 NREL-5MW wind turbines (Jonkman et al., 2009) modeled using SimWindFarm. The non-centralized strategy is compared with its centralized counterpart, showing a significant reduction in computation times while ensuring a good level of performance. The clustering performed in Changqing et al. (2022) allows the division of wind turbines into minimally coupled clusters. The approach is applied to a 25-turbine farm (1.5 MW), showing its effects on frequency regulation and power output w.r.t. more traditional control approaches. An HMPC scheme is proposed in Zhao et al. (2023), and the performance w.r.t. CMPC are qualitatively compared in a 12-turbine wind farm case study.

Transportation networks. The control approach in Chanfreut, Maestre, Hatanaka et al. (2022) is validated on an urban transportation network (De Oliveira & Camponogara, 2010) with eight intersections, and performance is compared w.r.t. CMPC. Simulations show how this strategy can reduce the number of active communication links more than the 40% while retaining good levels of performance

Mechanical systems. In Baldivieso Monasterios and Trodden (2021), a DMPC technique (Mayne et al., 2005) relying on tube-based MPC (Limon et al., 2010) is considered as the underlying control strategy for each coalition in a Coal-MPC scheme. The approach is successfully validated against CMPC over a four-agent mass–spring–damper planar chain, showing that the coalitional control scheme proposed can reach states that are otherwise not feasible for CMPC. In Chanfreut, Maestre, Ferramosca et al. (2022), coalitional control is used in combination with Dec-MPC. The approach proposed is validated over a 12-trucks system connected through springs and dampers; an example also used in Riverso and Ferrari-Trecate (2012) and Trodden and Maestre (2017), showing a good performance retention w.r.t. centralized control with significant reductions in communication costs. Validation of the approach in Masero, Baldivieso-Monasterios et al. (2023) is performed through the control of a four-trucks system in a coupled chain configuration as also tested in Trodden and Maestre (2017). A fifth truck is added during the simulation to show the plug-and-play capabilities.

Smart buildings. Few studies are present in this field, which is often used in the literature to develop robust and stochastic approaches. The approach developed in Zheng et al. (2018) is validated using a building temperature regulation problem against CMPC. The system comprises eight rooms that should keep the temperature variation at zero despite external influences. The DMPC approach can stabilize the network, as CMPC, but only qualitative results are provided, and some performance degradation is present. The paper also provides theorems for the stability and recursive feasibility of the DMPC strategy. On the contrary, the efficacy of the partitioning in Atam and Kerrigan (2021) is assessed through ad-hoc performance indicators for the specific application or zone temperature control. The approach is extensively validated for the Dec-MPC control of a 5- and a 20-zones case study, also considering results for exhaustive enumeration of the possible partitions, and compared with the partitioning approach of Chandan and Alleyne (2013).

Abstract networks. The partitioning approach proposed in Wang and Koeln (2023) is validated by computing the size of the resulting minimal robust positively invariant sets for different clustering procedures, showing how it outperforms other strategies in maximizing the sizes of the sets. The case study is a 43 agents flow system (Koeln & Alleyne, 2017). However, the impact of the proposed partitioning on the performance of the robust Dec-MPC strategy has not been explored in the work. The partitioning approach defined in Riccardi et al. (2025c) is applied for partitioning a modular network with 64 agents, and a random network of hybrid systems with 50 agents. The first case shows how varying the granularity α allows to retrieve modules at different aggregation levels, allowing a hierarchical clustering. In the second case, an ADMM-based DMPC approach (Summers & Lygeros, 2012) is deployed for network control. Different simulations are performed: one for CMPC, one for the conventional DMPC-ADMM with 50 agents, and three for partitionings obtained with varying levels of α. The simulation results show how the optimization-based partitioning DMPC controllers have a loss of performance w.r.t. CMPC below 0.3%, while the conventional DMPC-AMM approach with 50 agents has a loss of more than 12%. This performance advantage is paid in computation time, which is generally higher for partitioned system w.r.t. the conventional ADMM formulation. The approaches are also compared in terms of computational cost by calculating the core seconds for the simulations, i.e., the number of seconds necessary to compute the control action in parallel times the number of agents working in parallel. In this regard, optimization-based partitioning allows a computational cost in line with CMPC, while conventional DMPC-ADMM is at least 2.59 times more expensive. Regarding algorithmic partitioning, the simulation results show how the loss in performance is of an additional 1% w.r.t. the ones obtained with optimization-based partitioning. However, the computation times are comparable to the ones of conventional DMPC-ADMM with 50 agents (1.75 times slower), but having the smallest computational cost among all the approaches in terms of core seconds.

Railway networks. The approach in Kersbergen, van den Boom et al. (2016) is validated on the model of the Dutch railway network (Kersbergen, Rudan et al., 2016) against a CMPC implementation. The results show that the distributed implementation is up to 90% faster in computing the predictive control action w.r.t. CMPC with only marginal performance losses.

Telecommunication systems. The strategy proposed in Masero et al. (2020b) is applied to the case of a network of 37 base stations to optimize the number of served users and energy consumption. The approach is validated against the more traditional best-signal-level approach (Fletscher et al., 2019), and decentralized and CMPC. Results show significant improvement of all the predictive control strategies w.r.t. the traditional approach, where coalitional control is the closest to CMPC in terms of performance while reducing the communication burden.

Industrial plants. In Chen et al. (2020) the update of the input trajectories in the cooperative DMPC is performed through the Gauss-Jacobi distributed optimization method (Bertsekas & Tsitsiklis, 2015). Proofs of feasibility and stability of the overall architecture are provided. The approach is tested over a walking beam reheating furnace system and a six-area power system, and validated against the DMPC formulation of Venkat et al. (2005). In the tests, the hierarchical approach of Chen et al. (2020) shows the ability to reduce the communication burden, avoiding the transmission of unnecessary information while ensuring system performance.

11.3. Applications for community-detection-based methods

Application of the modularity-based partitioning methodology derived in Jogwar and Daoutidis (2017) is performed in Moharir et al. (2018) for iterative DMPC of an Amine gas sweetening plant. The decomposition of the relatively small plant shows how modularity maximization is achieved when two communities are obtained, and further partitioning the system into three communities does not improve the modularity. Modularity maximization also accounts for the structural information about the plant, ensuring the existence of well-posed subsystems (i.e. subsystems for which a controller can be defined, having at least one input and one output of the original plant). No further division of the plant is proposed. The DMPC architecture is compared against CMPC, Dec-MPC, and DMPC for a different partitioning (sub-optimal in terms of modularity). The modularity-based DMPC is the best-performing non-centralized strategy, approaching CMPC results while reducing computation times. Given the reduced size of the

plant, all possible modularity-based partitions of the systems providing well-posed subsystems can be evaluated in this case; however, the procedure still relies on expert knowledge, heuristics, and inspection to be performed accurately.

The approach of Jogwar and Daoutidis (2017) is deployed in Pourkargar et al. (2019) for both distributed control and estimation of a benzene alkylation process consisting of four continuous stirred-tank reactors, and a flash tank separator. Deploying the DMPC architecture for the selected partition provides a good approximation of CMPC results with a reduced computation burden.

A modularity-based partitioning technique has been used in Guo et al. (2019) to deploy a DMPC strategy for perimeter control of urban traffic. The approach is structured to divide urban networks into regions for which traffic control methods based on the macroscopic fundamental diagram (Geroliminis & Daganzo, 2008) can be implemented (An et al., 2018). To this, a two-layer partitioning method is proposed in Guo et al. (2019). In the upper layer, congested regions are selected using the dynamic modularity metric for urban traffic introduced in Guo et al. (2019). These regions are compact, and a macroscopic fundamental diagram can be identified for them. However, the regions do not cover the entirety of the urban network, i.e. non-congested regions are present at their interconnection, defining a boundary. At the lower layer of the partitioning strategy, the boundary region is divided into multiple areas based on spatial proximity using the Euclidean distance, so that a boundary region exists between each two congested areas. Validation of the partitioning approach is performed by applying the DMPC strategy (Kim et al., 2019) on the case study of the road network in downtown Jinan, China. The proposed approach is validated against a fixed signal control rate, and the boundary-feedback control strategy (Zhu & Li, 2019), demonstrating how the proposed strategy is the most effective in reducing the total time spent on the road by the drivers, and the total accumulated delay of the vehicles.

Modularity optimization has been used in Wang et al. (2022) to partition a power network in the presence of photovoltaic inverters and electric vehicles, with the objective of using the charging/discharging capabilities of the latter to mitigate the curtailment of the former. In Wang et al. (2022), a two-step Dec-MPC strategy is developed: in the first phase a modified modularity index is used for partitioning, and in the second step local MPC actions are computed in parallel. The modularity metric is modified to incorporate two ad-hoc performance indicators for power networks. The first is voltage sensitivity, which describes how voltage magnitude changes in nodes after voltage injection in other nodes. The second is the voltage regulation capacity used for reactive power compensation. The modularity is maximized through the Louvain algorithm (Girvan & Newman, 2002). The resulting approach is qualitatively validated on the IEEE 123 node test feeder, showing the viability of the strategy.

The paper (He & Li, 2023) presents a graph-based hierarchical Lyapunov-based DMPC (Liu, Chen et al., 2010) framework. The control framework is based on the selection of communities performed through the multiway spectral community detection algorithm (Zhang & Newman, 2015). This community detection algorithm approaches the modularity maximization problem using spectral methods through a heuristic approach that can work with any number of desired communities. The approach has the same computational complexity of k-means clustering; therefore, it is attractive for its scalability. The method partitions subsystems into a relative leader-follower hierarchy by integrating community detection algorithms. The work is posed as an extension of Chen et al. (2020) to nonlinear systems. However, no formal guarantees are given, and the use of the interpretive structural modeling, as well as the communication strategy, are not entirely clear, contrary to its reference strategy. The proposed architecture minimizes all-to-all communication, requiring only a single inter-layer exchange per sample, reducing the computational burden. The approach is validated over a reactor-separator integrated system developed in Pourkargar et al. (2017).

Modularity-based algorithmic partitioning using iterative bisection (Newman, 2006) is also at the basis of the automatic decomposition approach used in the Shell-Yokogawa platform for advanced Control and estimation (Tang et al., 2023). In this advanced process control technology, partitioning is performed using an equivalent graph representation of the network, with the usual definition of nodes as variables and arcs as relations. Iterative bisection is performed according to the algorithm of Newman (2006), and the resolution parameter (Reichardt & Bornholdt, 2006) is used to limit the size of the resulting clusters. Two post-processing procedures are used to ensure the connectedness of the resulting components, and to re-balance the sets according to their sizes. Heuristics are used to define the number of clusters, and resolution. The partitioning algorithm is applied to three case studies: a crude distillation process for a refinery, a gas-to-liquid process, and a hydrocracking process, all plants with hundreds of nodes. The resulting partitions are used for the application of DMPC showing how the distributed computation of the control action can improve the time required for online optimization up to 5 times. However, the impact on the control performance of this approach w.r.t. CMPC is not assessed.

11.4. Applications for game-theoretic oriented methods

In this section, we report applications of the coalitional predictive control schemes discussed above to case studies that have not been presented already, specifically: the control of irrigation canal, freeway transportation, vehicle platooning, and cyber–physical systems.

The first known contribution in coalitional predictive control is Fele et al. (2014), where the problem of controlling an irrigation canal is addressed. The aim of the strategy is to optimize water distribution by dynamically adjusting coalitions of control agents to balance control performance and communication cost. The framework is hierarchical: in the top layer, the partition of the system into coalition is achieved through topology optimization, where the optimal topology is selected from a predefined set of possible topologies. Decentralized feedback gains are associated with each topology, and the solution of an LMI problem guides the partition selection. Then, at a lower level, Dec-MPC is applied. Coalition formation and local optimization work at different time scales. The control methodology is validated through the SOBEK hydrodynamic simulator (SOBEK, 2000) on a model of the Dez irrigation canal (Isapoor et al., 2011), and compared against CMPC showing suboptimal but adequate performance, without the need of a complete communication topology.

A hierarchical formulation of coalitional predictive control has also been applied to nonlinear systems in Chanfreut et al. (2021a). In particular, this study focuses on freeway traffic control through ramp metering and variable speed limits (Papageorgiou et al., 2008; Papageorgiou & Kotsialos, 2002). The solution proposed in Chanfreut et al. (2021a) consists of a two-level structure: a top layer forms the coalitions, and at the bottom level, a DMPC strategy is deployed for the resulting coalitions, specifically feasible cooperation-based MPC (Venkat et al., 2008) with Genetic Algorithm solver (GA) (Goldberg, 1989). Moreover, the two layers operate at different time scales, with the top one being slower, allowing more time to solve the coalition formation problem. The study proposes as a potential solution to the coalition formation the bargaining procedure based on the Shapley value (Fele et al., 2017; Muros et al., 2018), or the PageRank method (Maestre & Ishii, 2017). To simplify the problem, only a limited set of possible coalitions is considered. The approach is extensively validated against Dec-MPC, and feasible cooperation-based MPC on a 15 km freeway segment, with multiple ramps, and speed-limiting devices. The results show a reduction in communication and coordination costs.

An application of coalitional predictive control to cyber–physical systems (Ding et al., 2021; Lee, 2015) with chain architecture is proposed in Maxim and Caruntu (2021). The key feature of this architecture is that the system first operates according to the non-cooperative

DMPC strategy (Scattolini, 2009), and when the feasibility of the solution fails, the system will switch to the coalitional predictive control formulation (Maxim et al., 2018). The switch occurs in cascade, triggered by one agent and propagating to its neighbors. Here, coalition formation is purely aggregative. The procedure is applied to a four-agents system, showing that when the local feasibility of non-cooperative DMPC is lost, then the application of coalitional predictive control can still provide satisfactory performance.

Vehicle platooning is the application considered in Maxim and Caruntu (2022) for the robust coalitional control strategy of Maxim and Caruntu (2021). The approach is tested on a four-car platoon detailed in Zhu et al. (2020), and string stability analysis (Dunbar & Caveney, 2012) is performed. The simulation shows how dynamic coalition formation stabilizes the platoon's operation with reduced communication. The work (Maxim et al., 2024) is proposed as an alternative approach to Maxim and Caruntu (2021, 2022) for coalitional control of vehicle platoons, distinguishing itself by the ability of individual agents to aggregate into coalitions autonomously. This objective is achieved by periodical evaluation of the string stability index (Dunbar & Caveney, 2012). The approach is validated on a four identical vehicles platoon under three different testing conditions. The results show that an inversely proportional relationship exists between performance and string stability.

An eight-tank process is used as a case study to perform a comparative performance analysis between DMPC and coalitional control in Maxim et al. (2023). In the paper, two non-cooperative DMPC formulations, one using a state-space model and the other an input–output model, are used to validate the performance of the coalitional control strategy based on a matrix gain feedback controller obtained through a gradient-based optimization previously introduced in Maxim et al. (2022). The Coal-MPC methodology allows the switch between decentralized and distributed communication topologies according to performance satisfaction. This switching Coal-MPC method shows results that are comparable with the non-cooperative DMPC strategy while allowing for a reduction in the communication burden.

12. Conclusions and future work

12.1. Contributions

This survey presents the first systematic classification and in-depth analysis of partitioning techniques for non-centralized predictive control. The scope of this work is both to unify the approaches currently present in the literature under a single framework, and to lay solid methodological foundations for future developments.

These objectives are achieved through the novel contributions of this work, which we summarize in the following. First, we introduce a formal reformulation of the partitioning problem in terms of mixed-integer programming, showing how, in the context of predictive control, the problem requires the solution of a bi-level optimization program, where network control performance is the cost functional of the partitioning problem. This aspect is at the basis of the complexity of network partitioning for control. Developing this framework, we introduce the concept of predictive partitioning, which uses predicted topology behavior to obtain the optimal network partitioning over the prediction horizon. Given the inherent NP-hard nature of these problems, their optimization-based solution would be prohibitive in real time; therefore, developing such a framework using greedy or heuristic algorithms or data-driven approaches would be advisable. Moreover, we introduce the concept of multi-topological network representations, which can serve as a basis for applying partitioning methodologies on networks whose topology and dynamical coupling are driven by different factors, such as events, time, network dynamics, or stochastic phenomena. Additionally, we provide a systematization of the key performance indicators to assess the quality of a partitioning for network control. On this basis, we establish an evaluation methodology that allows the direct comparison of different partitioning strategies. Such an approach can be the basis of further systematic development in this field, providing solid quantitative metrics for performance assessment.

In addition, this survey proposes several other ways to analyze and organize the literature in partitioning for predictive control. We start by presenting a systematization of network equivalents based on graphs. Then we introduce a classification of the partitioning techniques based on five main classes: optimization-based, algorithmic, community-detection-based, game-theoretic-oriented, and heuristic partitioning. For each class we discuss its level of optimality, scalability, complexity of computation and implementation, technical requirements, and other specific features it might exert. Further we introduce a functional sub-classification of the partitioning techniques, introducing cross-methodological partitioning objectives. We conclude the survey by discussing the known applications of the partitioning techniques proposing, when possible, reference systems for further developments and comparison.

12.2. State of the field

From the extended assessment of the partitioning techniques for the application of non-centralized MPC control, it results that many fundamental approaches have now been established. Specifically, abstract representations of networks of systems or of optimization problems are now a solid foundation to abstract the partitioning problem into the domain of graph partitioning. Most techniques use graph representation as a starting point, and, in this sense, they mostly differ from the type of weighting used, more topology- or system-oriented. What is more unconventional is the use of graphs capturing the 'flow' of energy among the nodes of a graph, where this has to be interpreted as sequences of the state variables, measured or predicted. Regarding the partitioning methodologies, optimization-based approaches are appreciated for their expressive power in terms of problem formulation, but are limited in scalability due to the NP-hard nature of the problem. Consequently, most works in partitioning focus on deriving specialized algorithms. In this domain, most approaches have focused so far on static network topologies, and methods for time-varying graphs or for plug-and-play operations are still at the forefront of research. The limitations here are given by the online re-partition of the network, which is still prohibitive to be performed on the same time-scale as the control action. Regarding this last point, hierarchical approaches that work on a slower time scale provide a viable solution. Finally, most works still consider partitioning as a distinct feature w.r.t. the control method. This happens because partitioning is fundamentally a bi-level problem; therefore, fixing a partition allows us to find a practical workaround for study and implementation. Consequently, control properties of the non-centralized architecture that originate from different topological structures are rarely considered. Overall, the field of partitioning can be considered mature for static sub-optimal partitioning methods, and well-developed for small topological changes in the structure of the network. Instead, works for which the topology is subject to fast and extended changes, uncertainties, or disturbances are currently missing.

12.3. Future work

Regarding future work in the field of partitioning for non-centralized predictive control, we believe it should focus on addressing the aspects indicated in the next paragraphs, to reach a level of sophistication for the resulting strategies such that they can adapt online to topological changes while ensuring the stability of the network, the feasibility of the control actions, robustness with respect to unexpected events, and minimal losses in terms of global optimality.

Table A.9
Analytical classification table.

Work	Year	Control method	Partitioning method	Application
Ocampo-Martinez et al. (2011)	2011	H-Dec-MPC	Graph-partitioning-based ordering algorithm (GPB)	Barcelona DWN
Ocampo-Martinez et al. (2012)	2012	H-Dec-MPC	Nested epsilon decomposition	Barcelona DWN
Fele et al. (2014)	2014	H-Coal-MPC	Coalition formation based on topology optimization from a predefined set	Irrigation canal networks
Núñez et al. (2015)	2015	Dec-, D-, and H- MPC	MI optimization partitioning	16 tanks water system
Kamelian and Salahshoor (2015)	2015	Dec-NLin-MPC	Algorithmic partitioning	Two-reactor (CSTR) chain followed by a flash separator with recycle
Xie et al. (2016)	2016	DMPC	Genetic algorithm minimization of input–output coupling between subsystems	Chemical plant: Tennessee Eastman problem. Five operation units: a reactor, a condenser, a compressor, a separator, and a stripper.
Kersbergen, van den Boom et al. (2016)	2016	DMPC	MIQP optimization for constraints decomposition	Dutch railway network
Pourkargar et al. (2017)	2017	CMPC, iterative and sequential DMPC	Community detection through modularity maximization	Reactor-separator process
Fele et al. (2017)	2017	Coal-MPC	Game theoretic coalition formation based on Shapley value	Smart grids
Maestre and Ishii (2017)	2017	Coal-MPC	Coalition formation based on an algorithm to handle aid requests sorted using distributed PageRank	16 tanks water system
Fele et al. (2018)	2018	Coal-MPC	Coalition formation based on bargaining procedure and TU-games	Wide-area control of power grids
Zheng et al. (2018)	2018	Dual mode DMPC	Algorithmic partitioning based on coupling degree	Building thermal management: eight rooms
Tang, Pourkargar et al. (2018)	2018	DMPC (noncooperative and iterative)	Relative Time-Averaged Gain Array (RTAGA)-based algorithmic modularity maximization over weighted IO bipartite graph using fast unfold	Reactor-separator process: 2CSTRs
Tang, Allman et al. (2018)	2018	DMPC-ADMM for nonlinear systems	Community-based decomposition of the optimization problem based on bipartite and unipartite representations, and fast unfold algorithm	Reactor-separator process: 2CSTRs
Rocha et al. (2018)	2018	Linearized cooperative and non-cooperative DMPC for nonlinear systems	Algorithmic partitioning based on variables matching and controllability check	Reactor-separator process: 2CSTRs
Jain et al. (2018)	2018	Dec-MPC	Heuristic partitioning based on ad-hoc performance index (modal participation matrix)	Northeast Power Coordinating Council nonlinear power system model
Moharir et al. (2018)	2018	DMPC (iterative)	Modularity-based partitioning (iterative division)	Amine gas sweetening plant
Muros et al. (2018)	2018	Coal-MPC	Coalition formation based on estimation of Shapley value and randomized methods	Barcelona DWN
Zhang et al. (2019)	2019	Enhancing DMPC	Data-driven partitioning using k-Shape	Shanghai WDN
Ye et al. (2019)	2019	HMPC	Heuristic partitioning (optimization-based)	Modified IEEE One Area RTS-96 network with wind turbines
Liu et al. (2019)	2019	HMPC	Heuristic partitioning (algorithmic based on dominant and connecting clusters)	Four vehicles platoon
Pourkargar et al. (2019)	2019	DMPC	Modularity-based partitioning (iterative division)	Benzene alkylation process: four continuous stirred-tank reactors, and a flash tank separator
Barreiro-Gomez et al. (2019)	2019	DMPC based on density-dependent population games	Multiobjective optimization, computed through distributed algorithm for graph partitioning	Barcelona DWN
Guo et al. (2019)	2019	DMPC for perimeter control	Modularity-based partitioning based on dynamic traffic estimation	Road network in downtown Jinan, China
Chen et al. (2020)	2020	Cooperative DMPC, over a sequential hierarchical down-stream of solutions	Hierarchical interpretive structural modeling (ISM)	Walking beam reheating furnace system, six-area power system
Wei et al. (2020)	2020	DMPC (Cooperative)	Algorithmic partitioning based on threshold given by coupling sensitivity analysis	Four-tanks water systems

(*continued on next page*)

Table A.9 (*continued*).

Siniscalchi-Minna et al. (2020)	2020	H-NCen-MPC	MIP optimization using ad hoc indicator (wake effect)	42 turbines farm (NREL-5 MW)
Lin et al. (2020)	2020	HMPC	Frequency-based fuzzy c-means algorithmic partitioning	20 turbines farm (NREL-5 MW)
Masero et al. (2020b)	2020	Coal-MPC	Hierarchical time-varying	Next-generation cellular networks with 37 base stations
Baldivieso Monasterios and Trodden (2021)	2021	Coal-MPC	Coalition formation based on consensus optimization and potential games	Mass-spring-damper planar chain
Chanfreut et al. (2021a)	2021	H-Coal-MPC	Coalition formation based on bargaining procedure and TU-games, or PageRank method	Freeway transportation network, METANET model
Masero, Maestre et al. (2021)	2021	H-Coal-MPC	Coalition formation based TU-games, and mixed-integer selection of the coalitions with predicted topologies	Eight tanks water system
Maxim and Caruntu (2021)	2021	Coal-MPC and DMPC	Coalition formation based on cooperative game	Theoretical four agents chain system
Masero, Frejo et al. (2021)	2021	H-Coal-MPC	Loop-pair clustering	Parabolic-trough solar collector fields with 100 loops
Segovia et al. (2021)	2021	DMPC based on optimality condition decomposition (OCD)	Modularity-based partitioning of the optimization problem	Quadruple-tank benchmark; Barcelona DWN
Atam and Kerrigan (2021)	2021	Dec-MPC	MI optimization, robust and stochastic	5 and 20 zones thermal buildings
Ananduta and Ocampo-Martinez (2021)	2021	Dec-MPC for economic dispatch	Heuristic partitioning based on communication protocol (algorithmic)	PG&E 69-bus distribution network
Chanfreut, Maestre, Ferramosca et al. (2022)	2022	Coal-MPC and Dec-MPC	Coalition formation based on cooperative game and invariant sets	12 trucks system
Maxim and Caruntu (2022)	2022	Coal-MPC and DMPC	Coalition formation based on cooperative game	Autonomous vehicle platooning
Sánchez-Amores et al. (2022)	2022	Coal-MPC	Coalition formation based on private and public factors	8 tanks input-coupled water system
Masero et al. (2022)	2022	H-NLin-Coal-MPC	Market-based coalition formation strategy	Parabolic-trough solar collector fields with 100 loops
Wang et al. (2022)	2022	Dec-MPC	Modularity-based partitioning using ad-hoc performance indicators	IEEE 123 node test feeder
La Bella et al. (2022)	2022	HMPC	k-way partitioning using METIS on a flow graph	IEEE 118-bus
Changqing et al. (2022)	2022	HMPC	k-means clustering for wake-effect interaction minimization	25 turbines farm (1.5 MW)
Chanfreut, Maestre, Hatanaka et al. (2022)	2022	Dec-MPC	Binary quadratic programming (BQP)	Urban traffic network with 8 intersections
Chanfreut et al. (2023)	2023	DMPC, ADMM- or ALADIN-based	k-means clustering	Solar parabolic trough plants
He and Li (2023)	2023	Lyapunov-based DMPC	Hierarchical multiway spectral community detection	Reactor-separator process
Huanca et al. (2023)	2023	Distributed Switching MPC	Sphere packing clustering combined with MPC	Quadrotor UAV swarm control
Masero, Baldivieso-Monasterios et al. (2023)	2023	H-Coal-MPC with PnP capabilities	Coalition formation based on invariant sets and dynamic scaling factors	4 + 1 trucks system
Masero, Ruiz-Moreno et al. (2023)	2023	H-NLin-Coal-MPC based on neural networks	Neural-networks-based market-based coalition formation strategy	Parabolic-trough solar collector fields with 100 loops
Maxim et al. (2023)	2023	Coal-MPC with switching topologies	Coalition formation based on cooperative game	8 tanks water system
Sánchez-Amores, Chanfreut et al. (2023)	2023	Coal-MPC	Coalition formation based on private and public factors	8 tanks input-coupled water system
Sánchez-Amores, Martinez-Piazuelo et al. (2023)	2023	H-Coal-MPC	Arbitrary partitioning	Parabolic-trough solar collector fields with 100 loops
Wang et al. (2023)	2023	DMPC	Modularity-based partitioning using frequency metric, and gap metric	Reactor separator process (2CSTR and a flash separator); and air separation process
Changqing et al. (2022)	2023	HMPC	k-means clustering (crowd search) using a set of key performance indicators	12 turbines farm

(*continued on next page*)

Table A.9 (*continued*).

Wang and Koeln (2023)	2023	Dec-MPC	Agglomerative hierarchical clustering based on minimal robust positively invariant sets	43 agents flow-based network
Tang et al. (2023)	2023	DMPC	Modularity-based partitioning (iterative division)	Crude distillation process for a refinery, gas-to-liquid process, and a hydrocracking process
Maxim et al. (2024)	2024	Coal-MPC with switching topologies	Coalition formation based on string stability condition	Autonomous vehicle platooning
Arastou et al. (2025)	2025	DMPC	Algorithmic (Kernighan–Lin) partitioning using computational complexity metric	Richmond water distribution network; Barcelona DWN
Jogwar (2019)	2025	DMPC	Spectral community detection for modularity based on time-varying graph representation	Benzene alkylation process: 4CSTR, and a flash tank separator
Riccardi et al. (2025c)	2025	DMPC-ADMM for hybrid systems	Bi-level partitioning; algorithmic selection of system units, and algorithmic or optimization-based (BQP) partitioning; balancing intra- and inter-agent interactions, with granularity parameter	Modular network with 64 agents, random network of hybrid systems with 50 agents

Time-varying and predictive partitioning. Further practical and theoretical developments can be achieved in the field of time-varying partitioning approaches, especially considering predictive partitioning. Most works now focus on static topologies, but aspects such as component failures, reconfigurations, or operational mode shifts that can induce a change in the interconnections of the network are rarely accounted for. In addition, models for topology dynamics are absent in the literature. Having such models would be fundamental for the development of predictive partitioning techniques, which can proactively reconfigure the controllers to counteract topological changes.

Integration of data-driven and learning methods. Conventional partitioning approaches consider static and deterministic topologies that are known in advance. However, topological structure may, in practice, be driven by phenomena that can be hard to model but for which data is available, especially for infrastructures such as power transmission and traffic networks, which are often subject to recurrent or periodic operational modes. Data-driven approaches and learning methods can extract latent structure from the data available about the networks and predict topological changes. Accordingly, novel non-centralized predictive control architectures can be deployed to leverage such insights and improve the overall performance or resilience of the network.

Resilience, robustness, and security-aware partitioning. Non-centralized predictive control strategies are based on communication networks that can be susceptible to latency, packet loss, or malicious attacks. Future work should define partitioning methods that are resilient to network malfunction and attacks, maintaining control performance despite disruptions. This includes robust partition definitions that tolerate link failures, cyber–physical attacks, and asynchronous information updates, as well as partitioning strategies that explicitly incorporate security metrics into the design process.

Multi-objective and performance-driven partition criteria. Most partitioning approaches still focus on a single objective. However, real applications may require multi-objective criteria that balance control performance, communication overhead, computational load, and robustness. Future research should formalize composite metrics that reflect these trade-offs. Furthermore, feedback approaches that adapt the partition to improve the overall control performance of the network are missing in the literature.

Real-time adaptive partitioning. The partitioning problem is known to be computationally intensive, and, consequently, most approaches work offline or on a different time scale w.r.t. real-time control. A direction to explore includes real-time adaptive partitioning strategies that leverage domain-specific knowledge or heuristics to quickly adapt the partition of the network locally to small network changes, while maintaining high levels of performance or robustness of the network. On a slower time scale, global network re-partitioning can then find a new configuration to be applied at a later stage.

Theoretical guarantees and control properties. Considering the theoretical developments instead, only the framework of coalitional control currently offers solid guarantees of satisfying the properties of feasibility, stability, and robustness when partitioning is involved, with few studies addressing these issues in general. Therefore, such properties might be established for time-varying partitioning approaches under different non-centralized control frameworks.

Standardized evaluation and benchmarking. Currently, standard benchmarks for partitioning are missing in the literature. Such benchmarks can greatly accelerate the development of this field of advanced non-centralized control because they will allow research to quantify the improvements of novel techniques and validate them in concrete scenarios. A standard benchmark for partitioning should be a real-world-oriented system for which a model is available. The use of high-fidelity simulators of nonlinear systems would be a great addition to the benchmark. A fundamental characteristic is the presence of a centralized MPC controller that can serve as a reference approach. Data about control performance, computation time, computation cost, and communication cost should be made available. Additional standard approaches, such as decentralized or distributed MPC, can be helpful in further comparing novel approaches. It would be ideal to have such benchmarks for the main application fields of control, e.g. power, transportation, and chemical networks. The development of dedicated studies for testing the approaches on large-scale networks with more than 10000 agents would also be beneficial. Finally, the use of time-varying and model-driven topologies would also improve the studies. If implemented correctly, these benchmarks can be tested in the future for several partitioning strategies and provide standardized reference approaches.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Analytical classification table

In Table A.9, we report the references presenting the partitioning strategies that have been investigated throughout the survey. They are listed in chronological order, which allows us to further understand the order of development of the techniques. Additionally, we report

the control methodology that has been deployed in the study, essential details about the partitioning method developed, and the application considered for the validation of the overall architecture.

Data availability

Data reported in the survey is available in the long-term repository 4TU.ResearchData Riccardi et al. (2025b, 2026).

References

Afram, A., Janabi-Sharifi, F., Fung, A. S., & Raahemifar, K. (2017). Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system. *Energy and Buildings*, *141*, 96–113.

Alessio, A., & Bemporad, A. (2007). Decentralized model predictive control of constrained linear systems. In *2007 European control conference* (pp. 2813–2818).

Alvarado, I., Limon, D., Muñoz De La Peña, D., Maestre, J. M., Ridao, M. A., Scheu, H., Marquardt, W., Negenborn, R. R., De Schutter, B., Valencia, F., & Espinosa, J. (2011). A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *Journal of Process Control*, *21*, 800–815.

An, K., Chiu, Y., Hu, X., & Chen, X. (2018). A network partitioning algorithmic approach for macroscopic fundamental diagram-based hierarchical traffic network management. *IEEE Transactions on Intelligent Transportation Systems*, *19*, 1130–1139.

Ananduta, W., & Ocampo-Martinez, C. (2021). Event-triggered partitioning for non-centralized predictive-control-based economic dispatch of interconnected microgrids. *Automatica*, *132*, 1–8.

Annoni, J., Bay, C., Taylor, T., Pao, L., Fleming, P., & Johnson, K. (2018). Efficient optimization of large wind farms for real-time control. In *2018 annual American control conference* (pp. 6200–6205).

Apt, K. R., & Witzel, A. (2009). A generic approach to coalition formation. *International Game Theory Review*, *11*, 347–367.

Arastou, A., Wang, Y., & Weyer, E. (2025). Optimization-based network partitioning for distributed and decentralized control. *Journal of Process Control*, *146*, 1–10.

Atam, E., & Kerrigan, E. C. (2021). Optimal partitioning of multithermal zone buildings for decentralized control. *IEEE Transactions on Control of Network Systems*, *8*, 1540–1551.

Attri, R., Dev, N., & Sharma, V. (2013). Interpretive structural modelling (ISM) approach: An overview. *Research Journal of Management Sciences*, *2*, 3–8.

Auer, G., Giannini, V., Godor, I., Skillermark, P., Olsson, M., Imran, M. A., Sabella, D., Gonzalez, M. J., Desset, C., Blume, O., & Fehske, A. (2012). How much energy is needed to run a wireless network? *IEEE Wireless Communications*, *18*, 1–16.

Bakule, L. (2008). Decentralized control: An overview. *Annual Reviews in Control*, *32*, 87–98.

Baldivieso Monasterios, P. R., & Trodden, P. A. (2021). Coalitional predictive control: Consensus-based coalition forming with robust regulation. *Automatica*, *125*, 14.

Barber, M. J. (2007). Modularity and community detection in bipartite networks. *Physical Review E*, *76*, 1–9.

Barreiro-Gomez, J., Dörfler, F., & Tembine, H. (2018). Distributed robust population games with applications to optimal frequency control in power systems. In *2018 annual American control conference* (pp. 5762–5767).

Barreiro-Gomez, J., Ocampo-Martinez, C., & Quijano, N. (2019). Time-varying partitioning for predictive control design: Density-games approach. *Journal of Process Control*, *75*, 1–14.

Barreiro-Gomez, J., & Tembine, H. (2018). Constrained evolutionary games by using a mixture of imitation dynamics. *Automatica*, *97*, 254–262.

Bello, O., Abu-Mahfouz, A. M., Hamam, Y., Page, P. R., Adedeji, K. B., & Piller, O. (2019). Solving management problems in water distribution networks: A survey of approaches and mathematical models. *Water*, *11*, 1–29.

Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, *35*, 407–427.

Bertsekas, D. P. (1996). *Constrained optimization and Lagrange multiplier methods*. Athena Scientific.

Bertsekas, D. (1999). *Nonlinear programming* (2nd ed.). Athena Scientific.

Bertsekas, D. P., & Tsitsiklis, J. N. (2015). *Parallel and distributed computation: Numerical methods*. Athena Scientific.

Blizard, A., & Stockar, S. (2025). Optimality loss minimization in distributed control with application to district heating. http://dx.doi.org/10.48550/arXiv.2507.02144.

Blondel, V. D., Guillaume, J., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*, 1–12.

Bondy, J. A., & Murty, U. S. R. (2008). *Graph theory*. Springer.

Boukelia, T. e., & Mecibah, M.-S. (2013). Parabolic trough solar thermal power plant: Potential, and projects development in Algeria. *Renewable and Sustainable Energy Reviews*, *21*, 288–297.

Boulle, M. (2004). Compact mathematical formulation for graph partitioning. *Optimization and Engineering*, *5*, 315–333.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Brandes, U., Delling, D., Gaertler, M., Goerke, R., Hoefer, M., Nikoloski, Z., & Wagner, D. (2006). Maximizing modularity is hard. http://dx.doi.org/10.48550/arXiv.physics/0608255, *preprint*.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, *30*, 107–117.

Castro, J., Gómez, D., & Tejada, J. (2009). Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, *36*, 1726–1730.

Cesco, J. C. (1998). A convergent transfer scheme to the core of a TU-game. *Revista de Matemáticas Aplicadas*, *19*, 23–35.

Chakrabortty, A., & Khargonekar, P. P. (2013). Introduction to wide-area control of power systems. In *2013 American control conference* (pp. 6758–6770).

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

Chanfreut, P., Maestre, J. M., & Camacho, E. F. (2021a). Coalitional model predictive control on freeways traffic networks. *IEEE Transactions on Intelligent Transportation Systems*, *22*, 6772–6783.

Chanfreut, P., Maestre, J. M., & Camacho, E. F. (2021b). A survey on clustering methods for distributed and networked control systems. *Annual Reviews in Control*, *52*, 75–90.

Chanfreut, P., Maestre, J. M., Ferramosca, A., Muros, F. J., & Camacho, E. F. (2022). Distributed model predictive control for tracking: A coalitional clustering approach. *IEEE Transactions on Automatic Control*, *67*, 6873–6880.

Chanfreut, P., Maestre, J. M., Gallego, A. J., Annaswamy, A. M., & Camacho, E. F. (2023). Clustering-based model predictive control of solar parabolic trough plants. *Renewable Energy*, *216*, 10.

Chanfreut, P., Maestre, J. M., Hatanaka, T., & Camacho, E. F. (2022). Fast clustering for multi-agent model predictive control. *IEEE Transactions on Control of Network Systems*, *9*, 1544–1555.

Changqing, C., Xinran, L., & Hua, L. (2022). Frequency regulation control strategy of wind farms based on temporal and spatial uncertainty. *Sustainable Energy Technologies and Assessments*, *53*, 1–11.

Chen, Z., Liu, J., Lin, Z., & Duan, Z. (2019). Closed-loop active power control of wind farm based on frequency domain analysis. *Electric Power Systems Research*, *170*, 13–24.

Chen, M., Zhao, J., Xu, Z., Liu, Y., Zhu, Y., & Shao, Z. (2020). Cooperative distributed model predictive control based on topological hierarchy decomposition. *Control Engineering Practice*, *103*, 15.

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

Conejo, A. J. (Ed.), (2006). *Decomposition techniques in mathematical programming: Engineering and science applications*. Springer.

Conway, J. H., & Sloane, N. J. A. (1988). *Sphere packings, lattices and groups*. Springer.

Curzi, G., Modenini, D., & Tortora, P. (2020). Large constellations of small satellites: A survey of near future challenges and missions. *Aerospace*, *7*, 1–18.

De Oliveira, L. B., & Camponogara, E. (2010). Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C: Emerging Technologies*, *18*, 120–139.

Diestel, R. (2017). *Graph theory*: *vol. 173*, (5th ed.). Springer.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*, 269–271.

Ding, D., Han, Q., Ge, X., & Wang, J. (2021). Secure state estimation and control of cyber-physical systems: A survey. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*, 176–190.

(2015). *Distributed model predictive control with event-based communication*. Kassel Univiversity Press.

Dorri, A., Kanhere, S. S., & Jurdak, R. (2018). Multi-agent systems: A survey. *IEEE Access*, *6*, 28573–28593.

Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, *17*, 245–255.

Dunbar, W. B., & Caveney, D. S. (2012). Distributed receding horizon control of vehicle platoons: stability and string stability. *IEEE Transactions on Automatic Control*, *57*, 620–633.

Fele, F., Debada, E., Maestre, J. M., & Camacho, E. F. (2018). Coalitional control for self-organizing agents. *IEEE Transactions on Automatic Control*, *63*, 2883–2897.

Fele, F., Maestre, J. M., & Camacho, E. F. (2017). Coalitional control: Cooperative game theory and control. *IEEE Control Systems Magazine*, *37*, 53–69.

Fele, F., Maestre, J. M., Hashemy, S. M., Muñoz de la Peña, D., & Camacho, E. F. (2014). Coalitional model predictive control of an irrigation canal. *Journal of Process Control*, *24*, 314–325.

Fine, T. L. (2006). *Feedforward neural network methodology*. Springer.

Fletscher, L. A., Suárez, L. A., Grace, D., Peroni, C. V., & Maestre, J. M. (2019). Energy-aware resource management in heterogeneous cellular networks with hybrid energy sources. *IEEE Transactions on Network and Service Management*, *16*, 279–293.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, *486*, 75–174.

Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports*, *659*, 1–44.

Frank, S., & Rebennack, S. (2016). An introduction to optimal power flow: Theory, formulation, and examples. *IIE Transactions*, *48*, 1172–1197.

Gallego, A. J., & Camacho, E. F. (2012). Adaptative state-space model predictive control of a parabolic-trough field. *Control Engineering Practice*, *20*, 904–911.

Galloway, B., & Hancke, G. P. (2013). Introduction to industrial control networks. *IEEE Communications Surveys & Tutorials*, *15*, 860–880.

Gálvez-Carrillo, M., De Keyser, R., & Ionescu, C. (2009). Nonlinear predictive control with dead-time compensator: Application to a solar power plant. *Solar Energy*, *83*, 743–752.

Gauci, M., Chen, J., Li, W., Dodd, T. J., & Groß, R. (2014). Self-organized aggregation without computation. *The International Journal of Robotics Research*, *33*, 1145–1161.

Geroliminis, N., & Daganzo, C. F. (2008). Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings. *Transportation Research Part B: Methodological*, *42*, 759–770.

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, *99*, 7821–7826.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley.

Grigg, C., Wong, P., Albrecht, P., Allan, R., Bhavaraju, M., Billinton, R., Chen, Q., Fong, C., Haddad, S., Kuruganty, S., Li, W., Mukerji, R., Patton, D., Rau, N., Reppen, D., Schneider, A., Shahidehpour, M., & Singh, C. (1999). The IEEE reliability test system-1996. A report prepared by the reliability test system task force of the application of probability methods subcommittee. *IEEE Transactions on Power Systems*, *14*, 1010–1020.

Grunnet, J. D., Soltani, M., Knudsen, T., Kragelund, M., & Bak, T. (2010). Aeolus toolbox for dynamics wind farm model, simulation and control. In *European wind energy conference and exhibition* (pp. 1–10).

Guo, Y., Yang, L., Hao, S., & Gao, J. (2019). Dynamic identification of urban traffic congestion warning communities in heterogeneous networks. *Physica A. Statistical Mechanics and its Applications*, *522*, 98–111.

Gupta, A. (1996). Fast and effective algorithms for graph partitioning and sparse-matrix ordering. *IBM Journal of Research and Development*, *41*, 1–23.

He, W., & Li, S. (2023). Enhancing topological information of the Lyapunov-based distributed model predictive control design for large-scale nonlinear systems. *Asian Journal of Control*, *25*, 1476–1487.

Heemels, W. P. M. H., De Schutter, B., & Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, *37*, 1085–1091.

Heo, G., & Gader, P. (2010). An extension of global fuzzy c-means using kernel methods. In *International conference on fuzzy systems* (pp. 1–6).

Howlader, A. M., Senjyu, T., & Saber, A. Y. (2015). An integrated power smoothing control for a grid-interactive wind farm considering wake effects. *IEEE Systems Journal*, *9*, 954–965.

Huanca, C. P. E. Y., Incremona, G. P., & Colaneri, P. (2023). Design of a distributed switching model predictive control for quadrotor UAVs aggregation. *IEEE Control Systems Letters*, *7*, 2964–2969.

Isapoor, S., Montazar, A., Van Overloop, P. J., & Van De Giesen, N. (2011). Designing and evaluating control systems of the Dez main canal. *Irrigation and Drainage*, *60*, 70–79.

Ishii, H., & Tempo, R. (2010). Distributed randomized algorithms for the PageRank computation. *IEEE Transactions on Automatic Control*, *55*, 1987–2002.

Ishii, H., & Tempo, P. (2014). The PageRank problem, multiagent consensus, and web aggregation: A systems and control viewpoint. *IEEE Control Systems Magazine*, *34*, 34–53.

Jain, A., Chakrabortty, A., & Biyik, E. (2018). Distributed wide-area control of power system oscillations under communication and actuation constraints. *Control Engineering Practice*, *74*, 132–143.

Jamoom, M. B., Feron, E., & McConley, M. W. (1998). Optimal distributed actuator control grouping schemes. *vol. 2*, In *Proceedings of the 37th IEEE conference on decision and control* (pp. 1900–1905). vol.2.

Javid, Z., Kocar, I., Holderbaum, W., & Karaagac, U. (2024). Future distribution networks: A review. *Energies*, *17*, 1–45.

Jha, A. R. (2010). *Wind turbine technology*. CRC Press.

Jogwar, S. S. (2019). Distributed control architecture synthesis for integrated process networks through maximization of strength of input-output impact. *Journal of Process Control*, *83*, 77–87.

Jogwar, S. S., & Daoutidis, P. (2017). Community-based synthesis of distributed control architectures for integrated process networks. *Chemical Engineering Science*, *172*, 434–443.

Jonkman, J., Butterfield, S., Musial, W., & Scott, G. (2009). *Definition of a 5-MW reference wind turbine for offshore system development*: *Tech. rep. NREL/TP–500–38060, 947422*.

Kamel, M., Hamouda, W., & Youssef, A. (2016). Ultra-Dense networks: A survey. *IEEE Communications Surveys & Tutorials*, *18*, 2522–2545.

Kamelian, S., & Salahshoor, K. (2015). A novel graph-based partitioning algorithm for large-scale dynamical systems. *International Journal of Systems Science*, *46*, 227–245.

Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations* (pp. 85–103). Plenum Press.

Karypis, G., & Kumar, V. (1998). *METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*: *Tech. rep.*, University of Minnesota, Department of Computer Science.

Kernighan, B. W., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, *49*, 291–307.

Kersbergen, B., Rudan, J., van den Boom, T., & De Schutter, D. (2016). Towards railway traffic management using switching Max-plus-linear systems: Structure analysis and rescheduling. *Discrete Event Dynamic Systems: Theory and Applications*, *26*, 183–223.

Kersbergen, B., van den Boom, T., & De Schutter, B. (2016). Distributed model predictive control for railway traffic management. *Transportation Research Part C: Emerging Technologies*, *68*, 462–489.

Khalil, H. (2002). *Nonlinear systems* (3rd ed.). Pearson.

Kim, S., Tak, S., Lee, D., & Yeo, H. (2019). Distributed model predictive approach for large-scale road network perimeter control. *Transportation Research Record: Journal of the Transportation Research Board*, *2673*, 515–527.

Koeln, J. P., & Alleyne, A. G. (2017). Stability of decentralized model predictive control of graph-based power flow systems via passivity. *Automatica*, *82*, 29–34.

Kordestani, M., Safavi, A. A., & Saif, M. (2021). Recent survey of large-scale systems: Architectures, controller strategies, and industrial applications. *IEEE Systems Journal*, *15*, 5440–5453.

Kravaris, C., & Kantor, J. C. (1990). Geometric methods for nonlinear process control. 1. Background. *Industrial & Engineering Chemistry Research*, *29*, 2295–2402.

Kundur, P. S., & Malik, O. P. (2022). *Power system stability and control* (2nd ed.). McGraw Hill.

La Bella, A., Bonassi, F., Farina, M., & Scattolini, R. (2019). Two-layer model predictive control of systems with independent dynamics and shared control resources. *IFAC-PapersOnLine*, *52*, 96–101.

La Bella, A., Klaus, P., Ferrari-Trecate, G., & Scattolini, R. (2022). Supervised model predictive control of large-scale electricity networks via clustering methods. *Optimal Control Applications and Methods*, *43*, 44–64.

Lakshmi, K., Visalakshi, N. K., & Shanthi, S. (2018). Data clustering using K-Means based on crow search algorithm. *Sādhanā*, *43*, 1–12.

Larsen, G. H., Van Foreest, N. D., & Scherpen, J. M. A. (2014). Power supply–demand balance in a smart grid: An information sharing model for a market mechanism. *Applied Mathematical Modelling*, *38*, 3350–3360.

Lee, E. (2015). The past, present and future of cyber-physical systems: A focus on models. *Sensors*, *15*, 4837–4869.

Leicht, E. A., & Newman, M. E. J. (2008). Community structure in directed networks. *Physical Review Letters*, *100*, 5.

Limon, D., Alvarado, I., Alamo, T., & Camacho, E. F. (2010). Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, *20*, 248–260.

Lin, C. (1974). Structural controllability. *IEEE Transactions on Automatic Control*, *19*, 201–208.

Lin, Z., Chen, Z., Qu, C., Guo, Y., Liu, J., & Wu, Q. (2020). A hierarchical clustering-based optimization strategy for active power dispatch of large-scale wind farm. *International Journal of Electrical Power & Energy Systems*, *121*, 1–15.

Liu, J., Chen, X., Muñoz de la Peña, D., & Christofides, P. D. (2010). Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE Journal*, *56*, 2137–2149.

Liu, P., Kurt, A., & Ozguner, U. (2019). Distributed model predictive control for cooperative and flexible vehicle platooning. *IEEE Transactions on Control Systems Technology*, *27*, 1115–1128.

Liu, J., Muñoz De La Peña, D., Ohran, B. J., Christofides, P. D., & Davis, J. F. (2010). A two-tier control architecture for nonlinear process systems with continuous/asynchronous feedback. *International Journal of Control*, *83*, 257–272.

Liu, J., Muñoz De La Peña, D., & Christofides, P. D. (2009). Distributed model predictive control of nonlinear process systems. *AIChE Journal*, *55*, 1171–1184.

Louf, R., Roth, C., & Barthelemy, M. (2014). Scaling in transportation networks. *PLoS One*, *9*, 1–8.

Luan, X., De Schutter, B., Meng, L., & Corman, F. (2020). Decomposition and distributed optimization of real-time traffic management for large-scale railway networks. *Transportation Research Part B: Methodological*, *141*, 72–97.

Lyman, P. R., & Georgakis, C. (1995). Plant-wide control of the Tennessee Eastman problem. *Computers & Chemical Engineering*, *19*, 321–331.

Machowski, J. (2002). *Predictive control with constraints*. Prentice Hall.

Maestre, J. M., & Ishii, H. (2017). A PageRank based coalitional control scheme. *International Journal of Control, Automation and Systems*, *15*, 1983–1990.

Maestre, J. M., Muros, F. J., Fele, F., & Camacho, E. F. (2015). An assessment of coalitional control in water systems. In *2015 European control conference* (pp. 3286–3291).

Maestre, J. M., & Negenborn, R. R. (Eds.), (2014). *Distributed model predictive control made easy*: *vol. 69*, Springer.

Maestre, J. M., & Ocampo-Martinez, C. (Eds.), (2025). *Control systems benchmarks*. Springer.

Maestre, J. M., Muñoz de la Peña, D., Jiménez Losada, A., Algaba, E., & Camacho, E. F. (2014). A coalitional control scheme with applications to cooperative game theory. *Optimal Control Applications and Methods*, *35*, 592–608.

Martinez-Piazuelo, J., Quijano, N., & Ocampo-Martinez, C. (2022). Nash equilibrium seeking in full-potential population games under capacity and migration constraints. *Automatica*, *141*, 1–8.

Masero, E., Baldivieso-Monasterios, P. R., Maestre, J. M., & Trodden, P. A. (2023). Robust coalitional model predictive control with plug-and-play capabilities. *Automatica*, *153*, 14.

Masero, E., Fletscher, L. A., & Maestre, J. M. (2020a). A coalitional model predictive control approach for heterogeneous cellular networks. In *European control conference* (pp. 448–453).

Masero, E., Fletscher, L. A., & Maestre, J. M. (2020b). A coalitional model predictive control for the energy efficiency of next-generation cellular networks. *Energies*, *13*, 6546.

Masero, E., Frejo, J. R. D., Maestre, J. M., & Camacho, E. F. (2021). A light clustering model predictive control approach to maximize thermal power in solar parabolic-trough plants. *Solar Energy*, *214*, 531–541.

Masero, E., Maestre, J. M., & Camacho, E. F. (2022). Market-based clustering of model predictive controllers for maximizing collected energy by parabolic-trough solar collector fields. *Applied Energy*, *306*, 1–12.

Masero, E., Maestre, J. M., Ferramosca, A., Francisco, M., & Camacho, E. F. (2021). Robust coalitional model predictive control with predicted topology transitions. *IEEE Transactions on Control of Network Systems*, *8*, 1869–1880.

Masero, E., Ruiz-Moreno, S., Frejo, J. R. D., Maestre, J. M., & Camacho, E. F. (2023). A fast implementation of coalitional model predictive controllers based on machine learning: Application to solar power plants. *Engineering Applications of Artificial Intelligence*, *118*, 1–10.

Maxim, A., & Caruntu, C. (2021). A coalitional distributed model predictive control perspective for a cyber-physical multi-agent application. *Sensors*, *21*, 1–16.

Maxim, A., & Caruntu, C. (2022). Coalitional distributed model predictive control strategy for vehicle platooning applications. *Sensors*, *22*, 1–20.

Maxim, A., Maestre, J. M., Caruntu, C. F., & Lazar, C. (2018). Robust coalitional distributed model predictive control algorithm with stability via terminal constraint. In *2018 IEEE conference on control technology and applications* (pp. 964–969).

Maxim, A., Pauca, O., & Caruntu, C. (2023). Distributed model predictive control and coalitional control strategies - comparative performance analysis using an eight-tank process case study. *Actuators*, *12*, 1–23.

Maxim, A., Pauca, O., & Caruntu, C. (2024). Coalitional distributed model predictive control strategy with switching topologies for multi-agent systems. *Electronics*, *13*, 18.

Maxim, A., Pauca, O., Maestre, J. M., & Caruntu, C. F. (2022). Assessment of computation methods for coalitional feedback controllers. In *2022 European control conference* (pp. 1472–1477).

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*, 789–814.

Mayne, D. Q., Seron, M. M., & Raković, S. V. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, *41*, 219–224.

McAvov, T. J., & Ye, N. (1994). Base control for the Tennessee Eastman problem. *Computers & Chemical Engineering*, *18*, 383–413.

Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, *36*, 30–44.

Messmer, A., & Papageorgiou, M. (1990). METANET: A macroscopic simulation program for motorway networks. *Traffic Engineering & Control*, *31*, 466–470.

Metzger, M., & Polakow, G. (2011). A survey on applications of agent technology in industrial process control. *IEEE Transactions on Industrial Informatics*, *7*, 570–581.

Moharir, M., Pourkargar, D. B., Almansoori, A., & Daoutidis, P. (2018). Distributed model predictive control of an amine gas sweetening plant. *Industrial & Engineering Chemistry Research*, *57*, 13103–13115.

Monderer, D., & Shapley, L. S. (1996). Potential games. *Games and Economic Behavior*, *14*, 124–143.

Muros, F. J., Algaba, E., Maestre, J. M., & Camacho, E. F. (2017). The Banzhaf value as a design tool in coalitional control. *Systems & Control Letters*, *104*, 21–30.

Muros, F. J., Maestre, J. M., Ocampo-Martinez, C., Algaba, E., & Camacho, E. F. (2018). A game theoretical randomized method for large-scale systems partitioning. *IEEE Access*, *6*, 42245–42263.

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, *103*, 8577–8582.

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*, 1–15.

Núñez, A., Ocampo-Martinez, C., Maestre, J. M., & De Schutter, B. (2015). Time-varying scheme for noncentralized model predictive control of large-scale systems. *Mathematical Problems in Engineering*, *2015*, 1–17.

Ocampo-Martinez, C., Barcelli, D., Puig, V., & Bemporad, A. (2012). Hierarchical and decentralised model predictive control of drinking water networks: Application to Barcelona case study. *IET Control Theory & Applications*, *6*, 62–71.

Ocampo-Martinez, C., Bovo, S., & Puig, V. (2011). Partitioning approach oriented to the decentralised predictive control of large-scale systems. *Journal of Process Control*, *21*, 775–786.

Ocampo-Martinez, C., Puig, V., Cembrano, G., Creus, R., & Minoves, M. (2009). Improving water management efficiency by using optimization-based control strategies: the Barcelona case study. *Water Supply*, *9*, 565–575.

Ogata, K. (2022). *Modern control engineering* (5th ed.). Prentice Hall.

Paauw, J., Roossien, B., Aries, M. B. C., & Guerra Santin, O. (2009). Energy pattern generator - Understanding the effect of user behaviour on energy systems. In *First European conference energy efficiency and behaviour*.

Papageorgiou, M., Kosmatopoulos, E., & Papamichail, I. (2008). Effects of variable speed limits on motorway traffic flow. *Transportation Research Record: Journal of the Transportation Research Board*, *2047*, 37–48.

Papageorgiou, M., & Kotsialos, A. (2002). Freeway ramp metering: An overview. *IEEE Transactions on Intelligent Transportation Systems*, *3*, 271–281.

Paparrizos, J., & Gravano, L. (2015). k-Shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data* (pp. 1855–1870).

Pedroso, L., Batista, P., & Heemels, W. P. M. H. (2025). Distributed design of ultra large-scale control systems: Progress, challenges, and prospects. *Annual Reviews in Control*, *59*, 1–22.

Poullikkas, A. (2013). A comparative overview of large-scale battery systems for electricity storage. *Renewable and Sustainable Energy Reviews*, *27*, 778–788.

Pourkargar, D. B., Almansoori, A., & Daoutidis, P. (2017). Impact of decomposition on distributed model predictive control: A process network case study. *Industrial & Engineering Chemistry Research*, *56*, 9606–9616.

Pourkargar, D. B., Moharir, M., Almansoori, A., & Daoutidis, P. (2019). Distributed estimation and nonlinear model predictive control using community detection. *Industrial & Engineering Chemistry Research*, *58*, 13495–13507.

Rakhshani, E., Rouzbehi, K., J. Sánchez, A., Tobar, A. C., & Pouresmaeil, E. (2019). Integration of large scale PV-based generation into power systems: A survey. *Energies*, *12*, 1–19.

Rawlings, J. B., Mayne, D. Q., & Diehl, M. (2017). *Model predictive control: Theory, computation, and design* (2nd ed.). Nob Hill Publishing.

Ray, D. (2007). *A game-theoretic perspective on coalition formation*. Oxford University Press.

Reichardt, J., & Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, *74*.

Riccardi, A., Laurenti, L., & De Schutter, B. (2024a). Code underlying the publication: A benchmark for the application of distributed control techniques to the electricity network of the European economic area. http://dx.doi.org/10.4121/d2c0d075-1c49-41af-8113-5e50c27ca97e.

Riccardi, A., Laurenti, L., & De Schutter, B. (2024b). Code underlying the publication: A generalized partitioning strategy for distributed control. http://dx.doi.org/10.4121/90ada13d-a6c9-4e4c-a046-2b984595bcdd.v1.

Riccardi, A., Laurenti, L., & De Schutter, B. (2024c). A generalized partitioning strategy for distributed control. In *2024 IEEE 63rd conference on decision and control* (pp. 6134–6141).

Riccardi, A., Laurenti, L., & De Schutter, B. (2025a). A benchmark for the application of distributed control techniques to the electricity network of the European economic area. In *Control systems benchmarks* (pp. 1–20). Springer.

Riccardi, A., Laurenti, L., & De Schutter, B. (2025b). Code for publication: A general partitioning strategy for non-centralized control. http://dx.doi.org/10.4121/0e7dd651-66d7-451e-889b-d558e7d5b986.

Riccardi, A., Laurenti, L., & De Schutter, B. (2025c). A general partitioning strategy for non-centralized control. http://dx.doi.org/10.48550/arXiv.2502.21126, *preprint*.

Riccardi, A., Laurenti, L., & De Schutter, B. (2026). Code for publication: Partitioning techniques for non-centralized predictive control: A systematic review and novel theoretical insights. http://dx.doi.org/10.4121/f5e9c734-c14e-40f7-93a8-82884a0efbfc.

Riverso, S., & Ferrari-Trecate, G. (2012). Tube-based distributed control of linear constrained systems. *Automatica*, *48*, 2860–2865.

Rocha, R. R., Oliveira-Lopes, L. C., & Christofides, P. D. (2018). Partitioning for distributed model predictive control of nonlinear processes. *Chemical Engineering Research and Design*, *139*, 116–135.

Rogers, G. (2000). *Power system oscillations*. Springer.

Sánchez-Amores, A., Chanfreut, P., Maestre, J. M., & Camacho, E. F. (2022). Coalitional model predictive control with different inter-agent interaction modes. *European Journal of Control*, *68*, 1–7.

Sánchez-Amores, A., Chanfreut, P., Maestre, J. M., & Camacho, E. F. (2023). Robust coalitional model predictive control with negotiation of mutual interactions. *Journal of Process Control*, *123*, 64–75.

Sánchez-Amores, A., Martinez-Piazuelo, J., Maestre, J. M., Ocampo-Martinez, C., Camacho, E. F., & Quijano, N. (2023). Coalitional model predictive control of parabolic-trough solar collector fields with population-dynamics assistance. *Applied Energy*, *334*, 11.

Sandholm, W. H. (2010). *Population games and evolutionary dynamics*. The MIT Press.

Sandholm, T., Larson, K., Andersson, M., Shehory, O., & Tohmé, F. (1999). Coalition structure generation with worst case guarantees I. *Artificial Intelligence*, *111*, 209–238.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control – A review. *Journal of Process Control*, *19*, 723–731.

Segovia, P., Puig, V., Duviella, E., & Etienne, L. (2021). Distributed model predictive control using optimality condition decomposition and community detection. *Journal of Process Control*, *99*, 54–68.

Sezer, M. E., & Siljak, D. D. (1986). Nested e-decompositions and clustering of complex systems. *Automatica*, *22*, 321–331.

Shaferman, V., Schwegel, M., Glück, T., & Kugi, A. (2021). Continuous-time least-squares forgetting algorithms for indirect adaptive control. *European Journal of Control*, *62*, 105–112.

Shapley, L. (1953). A value for n-person games. In *Contributions to the theory of games II* (pp. 307–317). Princeton University Press.

Šiljak, D. D. (1991). *Decentralized control of complex systems*. Academic Press, Inc..

Šiljak, D. D. (2008). *Large-scale dynamic systems: stability and structure*. Dover Publications Inc..

Siniscalchi-Minna, S., Bianchi, F. D., Ocampo-Martinez, C., Domínguez-García, J. L., & De Schutter, B. (2020). A non-centralized predictive control strategy for wind farm active power control: A wake-based partitioning approach. *Renewable Energy*, *150*, 656–669.

Siri, S., Pasquale, C., Sacone, S., & Ferrara, A. (2021). Freeway traffic control: A survey. *Automatica*, *130*, 1–25.

Siringoringo, R., & Jamaluddin, J. (2019). Initializing the fuzzy c-means cluster center with particle swarm optimization for sentiment clustering. *Journal of Physics: Conference Series*, *1361*, 1–6.

Skogestad, S., & Postlethwaite, I. (2001). *Multivariable feedback control: Analysis and design* (2nd ed.). John Wiley & Sons.

(2000). *SOBEK user manual*: *Tech. rep.*, Delft, The Netherlands: WL Delft Hydraulics.

Son, Y. S., Baldick, R., Lee, K., & Siddiqi, S. (2004). Short-term electricity market auction game analysis: Uniform and pay-as-bid pricing. *IEEE Transactions on Power Systems*, *19*, 1990–1998.

Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: A survey. *Computer*, *27*, 17–26.

Stearns, R. E. (1968). Convergent transfer schemes for N-person games. *Transactions of the American Mathematical Society*, *134*, 449–459.

Stewart, B. T., Venkat, A. N., Rawlings, J. B., Wright, S. J., & Pannocchia, G. (2010). Cooperative distributed model predictive control. *Systems & Control Letters*, *59*, 460–469.

Stoffel, K., & Belkoniene, A. (1999). Parallel k/h-means clustering for large data sets. In *Euro-Par'99 parallel processing* (pp. 1451–1454).

Summers, T. H., & Lygeros, J. (2012). Distributed model predictive consensus via the alternating direction method of multipliers. In *50th annual allerton conference on communication, control, and computing* (pp. 79–84).

Tabuada, P. (2009). *Verification and control of hybrid systems: A symbolic approach*. Springer.

Tang, W., Allman, A., Pourkargar, D. B., & Daoutidis, P. (2018). Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection. *Computers & Chemical Engineering*, *111*, 43–54.

Tang, W., Carrette, P., Cai, Y., Williamson, J. M., & Daoutidis, P. (2023). Automatic decomposition of large-scale industrial processes for distributed MPC on the Shell–Yokogawa platform for advanced control and estimation (PACE). *Computers & Chemical Engineering*, *178*, 1–8.

Tang, W., & Daoutidis, P. (2018). Network decomposition for distributed control through community detection in input-output bipartite graphs. *Journal of Process Control*, *64*, 7–14.

Tang, W., Pourkargar, D. B., & Daoutidis, P. (2018). Relative time-averaged gain array (RTAGA) for distributed control-oriented network decomposition. *AIChE Journal*, *64*, 1682–1690.

Trodden, P. A., & Maestre, J. M. (2017). Distributed predictive control with minimization of mutual disturbances. *Automatica*, *77*, 31–43.

Vallee, F., Brunieau, G., Pirlot, M., Deblecker, O., & Lobry, J. (2011). Optimal wind clustering methodology for adequacy evaluation in system generation studies using nonsequential Monte Carlo simulation. *IEEE Transactions on Power Systems*, *26*, 2173–2184.

van Zyl, J. E., Savic, D. A., & Walters, G. A. (2004). Operational optimization of water distribution systems using a hybrid genetic algorithm. *Journal of Water Resources Planning and Management*, *130*, 160–170.

Venkat, A. N. (2006). *Distributed model predictive control: theory and applications* (Ph.D. thesis), University of Wisconsin-Madison.

Venkat, A. N., Hiskens, I. A., Rawlings, J. B., & Wright, S. J. (2008). Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, *16*, 1192–1206.

Venkat, A. N., Rawlings, J. B., & Wright, S. J. (2005). Stability and optimality of distributed model predictive control. In *Proceedings of the 44th IEEE conference on decision and control* (pp. 6680–6685).

Wang, L., Dubey, A., Gebremedhin, A. H., Srivastava, A. K., & Schulz, N. (2022). MPC-based decentralized voltage control in power distribution systems with EV and PV coordination. *IEEE Transactions on Smart Grid*, *13*, 2908–2919.

Wang, W., & Koeln, J. P. (2023). Hierarchical clustering of constrained dynamic systems using robust positively invariant sets. *Automatica*, *147*, 1–8.

Wang, J., Song, C., Zhao, J., Mo, Z., & Xu, Z. (2023). Distributed model predictive control-oriented network decomposition based on full dynamic response. *AIChE Journal*, *69*, 1–15.

Wei, Y., Li, S., & Wu, J. (2020). Event-triggered distributed model predictive control with optimal network topology. *International Journal of Robust and Nonlinear Control*, *30*, 2186–2203.

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

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*, 645–678.

Ye, L., Zhang, C., Tang, Y., Zhong, W., Zhao, Y., Lu, P., Zhai, B., Lan, H., & Li, Z. (2019). Hierarchical model predictive control strategy based on dynamic active power dispatch for wind power cluster integration. *IEEE Transactions on Power Systems*, *34*, 4617–4629.

Yin, X., Liu, Y., Yang, L., & Gao, W. (2022). Abnormal data cleaning method for wind turbines based on constrained curve fitting. *Energies*, *15*, 6373.

Zecevic, A., & Siljak, D. D. (2010). *Control of complex systems: Structural constraints and uncertainty*. Springer.

Zhang, X., & Newman, M. E. J. (2015). Multiway spectral community detection in networks. *Physical Review E*, *92*, 1–8.

Zhang, Y., Zheng, Y., & Li, S. (2019). Enhancing cooperative distributed model predictive control for the water distribution networks pressure optimization. *Journal of Process Control*, *84*, 70–88.

Zhao, J., Zhang, T., Tang, S., Zhang, J., Zhu, Y., & Yan, J. (2023). Optimal scheduling strategy of wind farm active power based on distributed model predictive control. *Processes*, *11*, 1–22.

Zheng, Y., Wei, Y., & Li, S. (2018). Coupling degree clustering-based distributed model predictive control network design. *IEEE Transactions on Automation Science and Engineering*, *15*, 1749–1758.

Zhou, Y., Rao, B., & Wang, W. (2020). UAV swarm intelligence: Recent advances and future trends. *IEEE Access*, *8*, 183856–183878.

Zhu, Y., He, H., & Zhao, D. (2020). LMI-based synthesis of string-stable controller for cooperative adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, *21*, 4516–4525.

Zhu, W., & Li, S. (2019). Study on discrete boundary-feedback-control strategy for traffic flow based on macroscopic fundamental diagram. *Physica A. Statistical Mechanics and its Applications*, *523*, 1237–1247.