Delft University of Technology

Delft Center for Systems and Control

Technical report 24-022

A generalized partitioning strategy for distributed control*

A. Riccardi, L. Laurenti, and B. De Schutter

If you want to cite this report, please use the following reference instead:

A. Riccardi, L. Laurenti, and B. De Schutter, "A generalized partitioning strategy for distributed control," *Proceedings of the 63rd IEEE Conference on Decision and Control*, Milan, Italy, pp. 6134–6141, Dec. 2024. doi:10.1109/CDC56724.2024.10885848

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/24_022

A Generalized Partitioning Strategy for Distributed Control

Alessandro Riccardi, Luca Laurenti, and Bart De Schutter

***Abstract*— The partitioning problem is a key problem for distributed control techniques. The problem consists in the definition of the subnetworks of a dynamical system that can be considered as individual control agents in the distributed control approach. Despite its relevance and the different approaches proposed in the literature, no generalized technique to perform the partitioning of a network of dynamical systems is present yet. In this article, we introduce a general approach to partitioning for distributed control. This approach is composed by an algorithmic part selecting elementary subnetworks, and by an integer program, which aggregates the elementary components according to a global index. We empirically evaluated our approach on a distributed predictive control problem in the context of power systems, obtaining promising performances in terms of reduction of computation speed and resource cost, while retaining a good level of performance.**

I. INTRODUCTION

The development of a distributed control architecture for a network of dynamical systems requires the selection of smaller subnetworks to be controlled independently by local controllers [1]. Coordination among local controllers is then achieved through a communication protocol [2]. The selection of the subnetworks is referred to as the *partitioning problem* [3]. In the literature concerning the partitioning problem, one can distinguish between two main approaches: i) the *top-down* approach, where a monolithic system is decomposed into smaller components defining the subnetworks; ii) the *bottom-up* approach, where individual systems, already predefined, aggregate to form bigger control entities. Despite being a central problem of distributed control a generalized partitioning technique is not present in the literature yet. With generalized approach we are referring to a partitioning technique that can be applied to every type of control system according to well-defined rules. The absence of a general theory unifying all the approaches leads to a superposition of terminology and concepts. Moreover, the technical problem of the definition of the number of subnetworks constituting the partitioning is not addressed in general, and it is often solved by heuristics or assumptions.

A. Contribution

In this work, we propose a novel approach to partitioning based on integer optimization and a novel global network metric. To approach the gaps in the literature, we propose a systematization of the definitions used in partitioning, supported by an initial mathematical formalization of the framework for a systematic approach to partitioning. Then, we define an algorithm for the selection of the elementary and indivisible control units in a network, to which we will refer to as *atomic control agents*. These elementary units will aggregate to constitute the subnetworks of the partitioning. Moreover, we define a global network metric accounting for both information about the dynamics and topology of the network. According to this novel metric, we define a partitioning strategy addressing simultaneously the two problems of: i) defining the number of subnetworks and ii) of which elementary system must belong to each set. The novel metric allows us to define the partitioning problem as an integer program. To illustrate the partitioning strategy, we use it to define the subnetworks for the application of a Distributed Model Predictive Control (DMPC) technique [4]. We test the resulting control architecture on a power system called the European Economic Area Electricity Network benchmark EEA-ENB [5].

A. Riccardi, L. Laurenti, and B. De Schutter are with the Delft Center for Systems and Control (DCSC), Delft University of Technology, 2600 AA Delft, The Netherlands.

Corresponding author: A. Riccardi, a.riccardi@tudelft.nl

II. LITERATURE SURVEY

In the following we report some of the most common partitioning techniques used in the literature, classifying them according to two main approaches.

A. Top-Down Approaches

Among the most relevant approaches there are the partitioning techniques based on the use of the modularity metric [6], also referred to as community detection strategies [7], [8], [9]. Modularity is a metric involving both information about the state transition matrix of a linear dynamical system, and topological information about the network. The scope of modularity-based techniques is to maximize the modularity. Maximization of modularity is known to be an NP-hard problem [10], and efficient algorithms have been developed to obtain approximate solutions to the problem of finding a partitioning of a network maximizing the modularity metric. The structure of these algorithms is based on the following pattern: given a certain network, modularity is maximized through an exchange procedure of state nodes across these communities until no further improvement in modularity is obtained.

Often partitioning techniques are derived in specialized contexts for ad-hoc applications. This is the case of the technique proposed in [11], where the partitioning of a wind farm is performed based on the wake-effect affecting down-stream turbines in the farm. In [12], flow-based distribution systems are considered, and more specifically a water distribution network. [12] proposes a multi-criteria optimization approach to obtain the partitioning. However, the number of sets of the partitioning have to be specified a priori.

B. Bottom-Up Approaches

A different approach to the partitioning problem is sought in coalitional control [13], [14] where control theory and game theory are used in combination. Coalitional control can be classified as a bottom-up approach, where individual control systems are aggregated into coalitions. These elementary control systems cannot be further divided, and constitute the fundamental blocks of a coalition, and therefore they are called agents. Each coalition, which is a collection of agents, is associated with a performance index, which is a function of the aggregated state and input values of the coalition. The performance cost is interpreted as an economic index, i.e. a transferable utility that can be reallocated among the agents. According to an iterative procedure, agents are exchanged across the coalitions until no further optimization of the coalitional cost is possible. Coalitional control inherits the complexity of the general clustering problem, thus requiring the solution of an NP-hard problem. A heuristic solution has been developed using binary quadratic programming in [15], thus providing better scalability for large systems when applying coalitional control.

Some partitioning techniques produce a non-stationary definition of the subnetworks. This is the case in the approaches defined in [16], [17], where linear switching systems and event-driven systems are considered. In these cases, it is necessary to establish the conditions that trigger a re-partition of the network, and the associated procedures to perform it.

Other works explore the effect of different choices of partitioning on the control properties of the systems, as done for distributed tube-based MPC in [18].

III. PRELIMINARY CONCEPTS AND NOTATION

A preliminary concept required to approach partitioning techniques is the one of equation graph, or hypergraph, or network graph [1]. Consider a linear discrete-time dynamical system of the form:

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

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, and A, B are matrices of appropriate dimensions. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ associated with the system is defined as a set of nodes $\mathcal{V} = \{u_1, \dots, u_p, x_1, \dots, x_n\}$, and a set of edges $\mathcal{E} = \{(i,j), |, i,j \in \mathcal{V}\}$, where an edge (i,j) exists if and only if a relation between nodes i and j is defined by a nonzero entry of matrices A or B. Moreover, we define a set of control nodes $\mathcal{U} = \{u_1, \dots u_p\}$, and a set of state nodes $\mathcal{X} = \{x_1, \dots x_n\}$. Accordingly, the entries of matrix B can be seen as the weights of directed edges in $\mathcal{E}_B$ connecting nodes from set $\mathcal{U}$ to set $\mathcal{X}$, and the entries of matrix A as edges in $\mathcal{E}_A$ connecting nodes in $\mathcal{X}$. Thus, we can see a dynamical system also as composed of two graphs, denoted by $(\mathcal{U} \bigcup \mathcal{X}, \mathcal{E}_B)$ and $(\mathcal{X}, \mathcal{E}_A)$. We can indicate a dynamical system with the notation $\mathcal{S}_i = \{\mathcal{U}_i \bigcup \mathcal{X}_i, \mathcal{E}_{A_i} \bigcup \mathcal{E}_{B_i}\} = \{\mathcal{V}_i, \mathcal{E}_i\}$. For a given node i, the set of all nodes connected to it is called the neighborhood of i, and denoted by $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}_i\}$. If we define a subgraph $\mathcal{S}_i = (\mathcal{V}_i, \mathcal{E}_i)$, then the frontier of the subgraph is the set of nodes inside the subgraph that are connected to nodes outside the subgraph, and we denote it as $\mathcal{F}_i = \{i \in \mathcal{V}_i \mid (i,j) \in \mathcal{E}, j \in \mathcal{V} \setminus \mathcal{V}_i\}$. Once the graph associated with the system is defined, the partitioning problem is then converted into the problem of finding suitable subsets of nodes and edges defining subgraphs constituting the subnetworks.

IV. THE CONCEPTS OF CONTROL AGENT AND ATOMIC CONTROL AGENT

For a given dynamical system, the scope of a partitioning strategy is to select subsystems that can be considered as individual control units. This partitioning is generally obtained through specified metrics, topological or non-topological, and/or objectives, which might be related to the optimality of the control actions of individual control units, or other criteria.

In order to formally define a partitioning strategy that can have a general applicability for control systems, we introduce the concept of *control agent*:

Definition 1 (Control Agent): A control agent is a dynamical system whose inputs affect only the state dynamics of the agent itself. All the dynamic relations that the control agent has with other control agents occur through dynamical coupling among the states, or directed output-input connections.

In this paper, we consider linear discrete-time time-invariant systems of the form (1). From **Definition 1**, a control agent[1], indexed by i, will take the form:

$$\begin{aligned} x^{[i]}(k+1) &= A^{[i]}x^{[i]}(k) + B^{[i]}u^{[i]}(k) + w^{[i]}(k) \\ w^{[i]}(k) &= \textstyle\sum_{j \in \mathcal{N}_i} A^{[ij]}x^{[j]}(k) \end{aligned} \tag{2}$$

where $x^{[i]} \in \mathbb{R}^{n_i}$, $u^{[i]} \in \mathbb{R}^{p_i}$, and $A^{[i]}$, $B^{[i]}$ are matrices of appropriate dimensions; $\sum_i n_i = n$, $\sum_i p_i = p$; $\mathcal{N}_i$ represents the neighborhood of agent i, $i \notin \mathcal{N}_i$; and $A^{[ij]}$ are matrices representing how the evolution of the states $x_j \in \mathcal{N}_i$ affect the dynamics of state x_i. The signal $w_k^{[i]}$ is the coupling effect that agent i experiences with its neighboring states. How this signal is interpreted is often at the basis of the definition of a non-centralized control strategy: in decentralized control, w_i is regarded as a disturbance; in DMPC, w_i may be considered as a bounded disturbance in non-iterative strategies, or as a known predicted signal over which agreement is sought through communication in iterative strategies [2].

In the definition of a general partitioning strategy, a fundamental task is to find a collection of smallest control agents in size that together cover the whole system, where the size of the control agent is the number of nodes constituting

[1]Equation (2) defining our control agent is a slightly different version of this form is used to specify a general subsystem [3], where the term $w_k^{[i]}$ also contains the coupling through inputs $B^{[ij]}u_k^{[j]}$. This approach is more general, but not of practical use for distributed control. Instead, it is advisable to achieve a decomposition of the network such that terms $B^{[ij]}$ are equal to zero to avoid coupling through input signals among subsystems. This selection of subsystems is always possible, and in the worst case it will provide a single agent corresponding to the entire network.

it. We specify this concept through the definition of atomic control agent.

Definition 2 (Atomic Control Agent): Given a network of dynamical systems, an atomic control agent is the smallest control agent definable through any decomposition of the network.

Atomic control agents represent the smallest individual components in a network of control agents. If a bottom-up partitioning approach is used, the definition of control agents is usually provided, but these might not coincide with the atomic control agents. We can set up a verification procedure to ensure these fundamental units are atomic control agents, and then proceed with the partitioning. Instead, in top-down partitioning, the definition of the atomic control agents is not present. These considerations are at the basis of the definition of an algorithm for the selection of atomic control agents. We conclude this section with a proposition that allows the construction of control agents as the result of the aggregation of other control agents. This result guarantees that the optimization-based partitioning will produce control agents by aggregation of the atomic control agents.

Proposition 1: The aggregation of multiple control agents is a control agent itself.

Proof: The statement can be verified by construction. Consider two control agents, $\mathcal{S}_1 = (\mathcal{V}_1, \mathcal{E}_1) = (\mathcal{X}_1 \bigcup \mathcal{U}_1, \mathcal{E}_1)$, $\mathcal{S}_2 = (\mathcal{V}_2, \mathcal{E}_2) = (\mathcal{X}_2 \bigcup \mathcal{U}_2, \mathcal{E}_2)$. By definition of control agent i, for both $\mathcal{S}_1$ and $\mathcal{S}_2$ it holds that no edges are present from the set of nodes $\mathcal{U}$ to the set of states $\mathcal{X}_1$ or $\mathcal{X}_2$. Now we define the agent resulting from their aggregation as $(\mathcal{V}_1 \bigcup \mathcal{V}_2, \mathcal{E}_1 \bigcup \mathcal{E}_2 \bigcup \{(i,j),(j,i), i \in \mathcal{X}_1, j \in \mathcal{X}_2\})$. The aggregation merges the sets of edges, with the addition of the edges in $\mathcal{E}$ linking the two sets of nodes, but not present in the individual sets of edges. The operation does not add any edge connecting $\mathcal{U}_i$ to $\mathcal{X}_j$. Thus, the agent resulting from the aggregation is a control agent according to **Definition 1** since the operation preserves the characteristics required in the definition.

V. ALGORITHM FOR THE SELECTION OF ATOMIC CONTROL AGENTS

The selection of atomic control agents produces fundamental entities that should be considered in a control strategy without further subdivisions. In our generalized approach, we first select the atomic control agents, and then determine the partitioning of the network by solving an integer program based on a global metric. This procedure is only a preliminary part of the overall network partitioning strategy, which will be completed by the optimization-based partitioning problem of **Section VIII**. We remark that, while the selection of atomic control agents may not be required for networks in which the smallest possible agents are given, it is required for top-down approaches to select the control agents for the partitioning. Moreover, the selection algorithm can be useful in bottom-up approaches to verify that the elementary agents provided are atomic control agents. To define an algorithm to select the atomic control agents according to the **Definitions 1** and **2**, we will use the graph representation of the network. We will indicate the atomic agents as sets $\mathcal{A}_i$, for $i = 1, \ldots, N_{\text{Atomic}}$, where $N_{\text{Atomic}} \leq |\mathcal{U}|$, and the collection of all the atomic control agents as $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_{N_{\text{Atomic}}}\}$. Also, we will make use of the additional set $\mathcal{L} \subset \mathcal{X}$, indicating the state nodes that remain to be assigned. We will use the superscript $[k]$ to indicate a certain set during k-th iteration. The algorithm consists of three main steps that we present in the following, and that will be performed in a specific order illustrated in **Algorithm 1**:

1) *Identify the roots of the atomic control agents*. These roots consist of at least one input node and one state node directly connected by a link. If a state node is connected to two or more input nodes, then the entire group constitutes a root, and all are merged into the same atomic control agent. This procedure is performed on the graph $(\mathcal{U} \bigcup \mathcal{X}, \mathcal{E}_B)$. We can have a maximum number of roots equal to $N_{\text{Atomic}} \leq |\mathcal{U}|$, where $N_{\text{Atomic}} \leq |\mathcal{U}|$ is the maximum number of atomic control agents. The roots are the first components of the atomic control agents.
2) Perform the assignment of the state nodes not belonging to the roots for which a directed edge from the state nodes in the roots exists. We call this procedure *forward assignment*. Here the graph considered is $(\mathcal{X}, \mathcal{E}_A)$. For each state node in the roots, e.g. x_i, we scan the set of unassigned state nodes. We assign one free state, e.g. x_j, to the root of x_i if the element $|A_{ji}|$ is the maximum over the row $A_{j\cdot}$. Note that, in this case, we are considering the directed edge ϵ_{ji}, from x_i to x_j with weight A_{ji}, which is different from the edge ϵ_{ij}. We call this phase *forward assignment*, since we look at the states from the roots to the periphery of the graph.
3) Perform the *backward assignment*, from the nodes belonging to the periphery of the graph toward the direction of the roots. Following a similar procedure as above, given a state x_j for which a forward assignment is not possible, and a state x_i belonging to a certain atomic control agent, we assign x_j to the agent of x_i if $|A_{ij}|$ is the maximum over the column $A_{\cdot j}$. In this case we are considering the directed edge ϵ_{ji}.

Algorithm 1 Selection of the Atomic Control Agents

Part 1 - Selection of the roots
Given $(\mathcal{U} \bigcup \mathcal{X}, \mathcal{E}_B)$ perform step 1)
  $(\mathcal{U}, \mathcal{X}) \rightarrow (\mathcal{A}^{[0]}, \mathcal{L}^{[0]})$
Part 2 - Selection of the atomic control agentns
Given $(\mathcal{X}, \mathcal{E}_A)$
  $k = 0$
  while $\mathcal{L}^{[k]}$ not empty $\vee$ $\mathcal{L}^{[k]}$ connected:
    perform *forward assignment*: step 2)
      $(\mathcal{A}^{[k]}, \mathcal{L}^{[k]}) \rightarrow (\mathcal{A}^{[k+1]}, \mathcal{L}^{[k+1]})$
    perform *backward assignment*: step 3)
      $(\mathcal{A}^{[k+1]}, \mathcal{L}^{[k+1]}) \rightarrow (\mathcal{A}^{[k+2]}, \mathcal{L}^{[k+2]})$
    $k = k + 2$

Once atomic control agents have been selected, partitioning techniques for control strategies can be designed considering as the smallest control agents the sets $\mathcal{A}_i$. We have implemented the algorithm for the selection of atomic control agents described above, and made it available in the open source toolbox [19]. The toolbox is designed to perform the selection starting from any linear system description of the form (1) representing a complex network. An example of the application of the algorithm is presented in **Section IX**. We conclude this section with a remark: if outputs are also specified in the dynamics of the system, the procedure defined above still holds, but it is necessary to assign the output nodes to each agent in the same way as done for the state nodes.

VI. THE OPTIMAL PARTITIONING

Before proceeding with the definition of a partitioning algorithm, we want to specify some concepts that are of general relevance for distributed control strategies and partitioning techniques.

Definition 3 (Control Partition of a Network): The control partition $\mathcal{P}$ of a network is defined as the collection of m control agents:

$$\mathcal{P} = \{\mathcal{S}_1, \dots, \mathcal{S}_m\} \tag{3}$$

where control agents $\mathcal{S}_i$ are sets of atomic control agents $\mathcal{A}_j$. For each set $\mathcal{S}_i$ of the partition, we define the set of its neighboring nodes as $\mathcal{N}_i = \{j \in \mathcal{X} \setminus \mathcal{V}_i \mid (i,j) \in \mathcal{E}_{A_i}\}$.

The definition of optimal partitioning strategy in the context of optimization-based control is related to the cost function used to obtain the control action, which we denote by J. When comparing a centralized control action u_c with a distributed u_d alternative applied to the same network, the optimal control partition $\mathcal{P}^*$ would be the one minimizing the difference $J(x, u_\mathrm{d}) - J^*(x, u_\mathrm{c})$. However, this index can not be computed for the cases in which the centralized control action u_c requires an intractable computation time. Thus, this is a theoretical index that can only be used to benchmark small-scale test cases and not as a cost function to obtain the optimal partitioning directly.

VII. THE PARTITIONING INDEX

A. The Two Main Problems

According to the discussion above, what we would like to obtain is a partitioning strategy that using information about the dynamics, and possibly the state, of the system will result in a distributed control cost as close as possible to the centralized strategy cost. As an example, for linear systems with linear constraints, in a completely decentralized control setting, local optimizers will not account for the interaction with neighbors to retrieve the local control action. In the ideal case in which no interaction is present among agents, the sum of individual contributions will match the overall cost. The difference between the global cost and the sum of local costs will increase as the interaction among control agents strengthens. Thus, for decentralized MPC, it is intuitive to require a partitioning strategy that will lead to the definition of control agents that have the least possible interaction among them. A similar principle applies to distributed MPC when non-iterative algorithms are considered. A further extension of this principle will be the one of strengthening as much as possible the interaction among the atomic agents participating in the same set.

Another aspect to consider in designing a partitioning strategy is the definition of the number of sets constituting the partition. Often, this number is assumed to be fixed, obtained through heuristics, or with iterative evaluation of different numbers of sets. What we can do instead is to define an optimization problem that will try to minimize a global metric of the network. We select this metric as the ratio between the inter-agent interaction and the intra-agent interaction, simultaneously defining to which set of the partition an atomic agent belongs using binary variables.

B. The Metric

To merge the two previous objectives simultaneously we define a new metric called *partitioning index*. The scope is to obtain a simple and static metric for partitioning. A similar approach can be used to exploit other types of information about the system, or the cost function of the optimization-based controller. Assume that, on the basis of the atomic control agents, we have a certain partition, constituted by control agents according to **Definitions 1** and **2**. We denote this partition as $\mathcal{P} = \{\mathcal{S}_1, \dots, \mathcal{S}_l\}$, with $l \leq m$. Each set $\mathcal{S}_i$ is obtained as the grouping of atomic agents $\mathcal{A}_j$. As a consequence, each set $\mathcal{S}_i$ is associated with a graph $(\mathcal{S}_i, \mathcal{E}_{\mathcal{S}_i})$ where the set of nodes groups all the nodes of the atomic control agents, and the set of the edges accounts for both the edges inside and between atomic control agents in the same set.

For each control agent $\mathcal{S}_i$ we define two indices, the intra-agent interaction $W^{\text{intra}}_{\mathcal{S}_i}$, and the inter-agent interaction $W^{\text{inter}}_{\mathcal{S}_i}$, which are functions of the sets $\mathcal{S}_i$, where the sets are not specified yet at this stage.

Each $\mathcal{S}_i$ is a network control agent by **Proposition 1**, obtained as the union of atomic control agents. Thus, we denote the state and input matrices associated with $\mathcal{S}_i$ as $A_{\mathcal{S}_i}$ and $B_{\mathcal{S}_i}$. We define the intra-agent interaction as:

$$W^{\text{inter}}_{\mathcal{S}_i} = \frac{\left(\sum_{z \in \mathcal{F}_{\mathcal{S}_i}} \sum_{j \in \mathcal{N}_{\mathcal{S}_i}} A^2_{z,j}\right)^{\frac{1}{2}}}{\sum_{k=1}^{l} \left(\sum_{z \in \mathcal{F}_{\mathcal{S}_k}} \sum_{j \in \mathcal{N}_{\mathcal{S}_k}} A^2_{z,j}\right)^{\frac{1}{2}}} \tag{4}$$

where $\mathcal{F}_{\mathcal{S}_i}$ is the frontier of the control agent $\mathcal{S}_i$, i.e. the set of nodes connecting to nodes outside the control agent, and $A_{z,j}$ are the entries of matrix A corresponding to the weights of the edges connecting the frontier nodes of the control agent $\mathcal{S}_i$ to its neighbors belonging to other control agents. The second index is:

$$W^{\text{intra}}_{\mathcal{S}_i} = \frac{\|A_{\mathcal{S}_i}\|_\mathrm{F}}{\sum_{k=1}^{l} \|A_{\mathcal{S}_k}\|_\mathrm{F}} \tag{5}$$

where $\|\cdot\|_\mathrm{F}$ indicates the Frobenius norm. Both indices are normalized, and are in the interval $[0, 1]$. The normalization

provides numerical results that will be suitable for comparing the ratios between different partitions, with different numbers of control agents.

With W_i^{intra} and W_i^{inter} we can associate to the partition $\mathcal{P} = \{\mathcal{S}_1, \dots, \mathcal{S}_l\}$ the following *specialized partition index*:

$$p^{\text{idx}}(\mathcal{P}) = \sum_{i=1}^{l} \left(\frac{W_{\mathcal{S}_i}^{\text{inter}}}{W_{\mathcal{S}_i}^{\text{intra}}} \right)^2 \tag{6}$$

The term p^{idx} constitutes our metric to define the partition minimizing the ratio between inter- and intra-agents coupling. The key feature of p^{idx} is that it accounts for *global* information about the network, which makes it a global metric. This fact will allow us to solve the two problems in **Section VII-A** simultaneously, as specified later in **Section VIII** with the definition of the partitioning strategy.

VIII. THE PARTITIONING STRATEGY

The strategy that we use for partitioning is based on the solution of a mixed integer optimization problem using the global metric (6). We introduce binary variables $\gamma_{ij} \in \{0, 1\}$ to specify if an atomic control agent $\mathcal{A}_i$ belongs to a set $\mathcal{S}_j$:

$$\gamma_{ij} = 1 \iff \mathcal{A}_i \subset \mathcal{S}_j \tag{7}$$

We have in total a number of m^2 binary decision variables, since $i, j = 1, \dots, m$, where i is related to the number of atomic control agents, and j to the number of sets. The number of sets $\mathcal{S}_i$ that will be generated by the procedure is not known a priori but it can not exceed the number m of atomic control agents by definition. We solve this practical problem in the implementation allowing for empty sets in the partitioning optimization problem. To be precise, we define the general partition of the network as $\mathcal{P} = \{\mathcal{S}_1, \dots, \mathcal{S}_m\}$, where m is the number of atomic control agents. However, of the m sets in $\mathcal{P}$, a certain number $m - l$ may be empty sets, where l is the number of non-empty sets. All decision variables γ_{ij} are collected into the vector γ, and the partition of the network is now a function of this vector, i.e. $\mathcal{P}(\gamma)$. Then, we define the general mixed integer partitioning problem as:

$$\begin{aligned} \min_{\gamma} \quad & p^{\text{idx}}(\mathcal{P}(\gamma)) \\ \text{s.t.} \quad & \sum_{j=1}^{m} \gamma_{ij} = 1 \quad \forall i = 1, \dots, m \\ & \gamma_{ij} \in \{0, 1\} \end{aligned} \tag{8}$$

where the equality constraint expresses the condition that each atomic control agent must belong exactly to one set, avoiding multiple assignments. This condition also ensures that all m atomic control agents are assigned. It can happen that the assignment will result in some sets being empty as specified before. In this case, we can just retain the l non-empty resulting sets. To use the metric (6) in the formulation (8) it is sufficient to notice that both $W_{\mathcal{S}_i}^{\text{inter}}$ and $W_{\mathcal{S}_i}^{\text{intra}}$ are functions of γ.

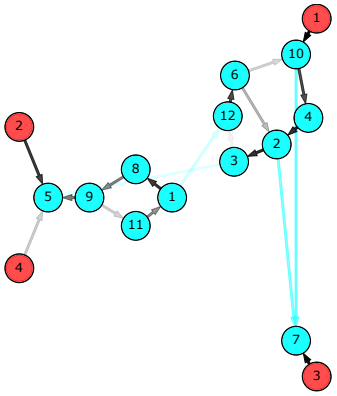

Fig. 1: Graph representation of the atomic control agents. Input nodes are in red, state nodes in blue. The nodes belonging to the same control agent are connected through black arrows. The interconnections among the control agents are represented by blue arrows. The transparency of the arrows represents the strength of interaction among nodes.

IX. EXAMPLE OF SELECTION OF ATOMIC CONTROL AGENTS

In this example, we consider a network in the form (1) with 4 inputs and 12 states. We have sparse matrices A and B, which we report in the following indicating the element in the i-th row and j-th column in the subscript:

$$\begin{array}{cccc} a_{1,11} = 0.5 & a_{2,4} = 0.98 & a_{2,6} = 0.31 & a_{3,2} = 0.97 \\ a_{4,10} = 0.8 & a_{5,3} = 0.11 & a_{5,9} = 0.6 & a_{6,12} = 0.84 \\ a_{7,2} = 0.54 & a_{7,10} = 0.58 & a_{8,1} = 0.92 & a_{9,8} = 0.53 \\ a_{10,4} = 0.36 & a_{10,6} = 0.18 & a_{10,10} = 0.02 & a_{11,9} = 0.2 \\ a_{12,1} = 0.17 & a_{12,3} = 0.06 & & \\ b_{5,2} = 0.65 & b_{5,4} = 0.15 & b_{7,3} = 0.84 & b_{10,1} = 0.87 \end{array} \tag{9}$$

We represent the network through a graph as indicated in **Section III**, thus having a set of input nodes $\mathcal{U} = \{u_1, \dots, u_4\}$, and of state nodes $\mathcal{X} = \{x_1, \dots, x_{12}\}$. No clear decomposition of this network can be established just by looking at the matrices. We can decompose this network by looking at its atomic control agents. Applying the algorithm presented in **Section IV** and implemented in the toolbox [19] we obtain the network in **Figure 1**, where input nodes are represented in red, and state nodes in blue. The selection procedure produces 3 atomic control agents from the original network, which has 4 inputs. Two inputs contribute to the same atomic control agent. Also, it is possible to see that two atomic control agents are not directly connected. The representation uses blue arrows to highlight the connection between atomic control agents. Once this selection is performed, distributed control strategies as well as further partitioning strategies can be applied to the network.

X. CASE STUDY: PARTITIONING FOR DISTRIBUTED PREDICTIVE CONTROL

A. System Description

In this section, we apply the partitioning strategy derived in this paper to define the subnetworks of an electrical

system for the application of a distributed control technique. The system considered is a benchmark for power networks called the European Economic Area Electricity Network Benchmark (EEA-ENB) [5]. It consists of 26 interconnected electrical areas, each representing a portion of the European power network. The topology of the network is illustrated in **Figure 2**.

Each electrical area is modeled as an equivalent electrical machine subject to a load request that has to be satisfied to maintain the network operating frequency of 50 [Hz] in the boundaries of ± 0.04 [Hz]. The problem of regulating the frequency deviation to zero under load variations is referred to as the Load Frequency Control (LFC) problem.

The benchmark includes data regarding the load demand, and the renewable energy generation for each area, which represent known external signals. The presence of renewable generation, with its intermittent and inertia-less characteristics, complicates the solution of the LFC problem. To help compensate for the presence of renewable generation, each electrical area is equipped with an energy storage system (ESS).

The dynamics of each electrical area is represented by the linear discrete-time system

$$\begin{aligned} \Delta\delta_i(k+1) &= \Delta\delta_i(k) + \tau 2\pi \Delta f_i(k) \\ \Delta f_i(k+1) &= \left(1 - \tfrac{\tau}{T_{\mathrm{p},i}}\right) \Delta f_i(k) + \tau \tfrac{K_{\mathrm{p},i}}{T_{\mathrm{p},i}} g_i(k) \\ e_i(k+1) &= e_i(k) + \tau \left(\eta_i^{\mathrm{c}} P_i^{\mathrm{ESS,\,c}}(k) - \tfrac{1}{\eta_i^{\mathrm{d}}} P_i^{\mathrm{ESS,\,d}}(k)\right) \end{aligned} \tag{10}$$

$$\begin{aligned} g_i(k) = \Delta P_i^{\mathrm{disp}}(k) - \Delta P_i^{\mathrm{load}}(k) + \Delta P_i^{\mathrm{ren}}(k) + \\ - \Delta P_i^{\mathrm{tie}}(k) - P_i^{\mathrm{ESS,\,c}}(k) + P_i^{\mathrm{ESS,\,d}}(k) \end{aligned} \tag{11}$$

$$\Delta P_i^{\mathrm{tie}}(k) = \sum_{j \in \mathcal{N}_i} T_{ij} (\Delta\delta_i(k) - \Delta\delta_j(k)), \tag{12}$$

where $\Delta\delta_i$, Δf_i, and e_i are respectively the power angle and frequency deviations from nominal conditions, and the energy stored in the ESS of the i-th electrical area. These variables represent the state of each area. The inputs of the system are $\Delta P_i^{\mathrm{disp}}$, $P_i^{\mathrm{ESS,\,c}}$, and $P_i^{\mathrm{ESS,\,d}}$, respectively the dispatchable power allocation of the i-th area, and the charging and discharging powers of the ESS. The signal $\Delta P_i^{\mathrm{load}}$ and $\Delta P_i^{\mathrm{ren}}$ are the load demand and the renewable energy production. The electrical areas exchange power according to $\Delta P_i^{\mathrm{tie}}$ representing the power transmission between the i-th and its neighbors $j \in \mathcal{N}_i$. The other symbols define parameters of the system used to model the network. A detailed explanation of their meaning and selection is present in [5], and the source code and documentation for the benchmark are present at [20]. Data for the load and renewable generation of the European network have been acquired from [21].

B. Distributed Control Strategy

To address the LFC control problem using distributed control, we implemented a distributed model predictive control scheme based on the alternating direction method of multipliers (DMPC-ADMM) [4], [22], [23]. This approach has proven to be suitable for controlling large-scale systems

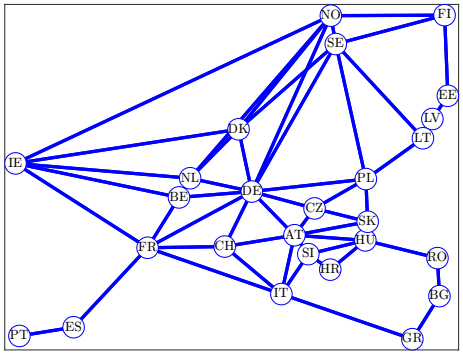

Fig. 2: Electrical topology of the EEA-ENB. Each node represents an electrical area, coinciding with a single country, whereas each edge is a transmission line.

where the application of centralized model predictive control (MPC) results unpractical, or not suited for real-time operation. In this case study we will compare the same DMPC-ADMM scheme implemented using different partitionings obtained with the strategy in **Section VIII**. The performance of the conventional DMPC-ADMM implementation differs only marginally from the one of a centralized MPC, which is in line with the literature [23]. In the DMPC-ADMM scheme, we select the parameter $\rho = 0.1$, as in [4].

C. Partitioning Strategy

In DMPC-ADMM it is usually assumed to have a network where control agents are already defined. In the following, we refer to this partitioning as $\mathcal{P}_{\mathrm{DMPC}}$. In this context, $\mathcal{P}_{\mathrm{DMPC}}$ will be our reference case for the other choices to compare performance, computation time, and computation resources, i.e. the number of [cores$\times$seconds] necessary to run the implementation in parallel.

The first approach we consider involves the specialized partitioning index defined in (6). We use this index as cost for the integer program (8). The integer program is solved using the genetic algorithm implementation of Matlab. The resulting partitioning $\mathcal{P}_{\mathrm{Opt},1}$ consists of 10 control agents.

For the second approach to partitioning, we use a modified version of the index (6). We add the term $1/(1 + |\bar{\mathcal{S}}_i|)$ to the cost of each control agent. Where $\bar{\mathcal{S}}_i$ is the system considered for the local DMPC optimization in ADMM. This system consists of the state and input variables of agent i and of all its direct neighbors. This choice aims to obtain the partitioning that simultaneously minimizes (6) and maximizes the size of the overlapping systems used in the local minimization of the stage cost for $\bar{\mathcal{S}}_i$. This choice will increase the complexity of local problems, because they will be larger, but will reduce the communication overhead, and possibly improve the agreement on shared variables among the control agents. We denote this partitioning as $\mathcal{P}_{\mathrm{Opt},2}$, and it has 8 control agents.

For an additional benchmark, we also consider a heuristic approach based on the geographical proximity of the atomic agents. We start from the consideration that the smallest previous partitioning is $\mathcal{P}_{\mathrm{Opt},2}$ with 8 control agents, thus

TABLE I: Comparison metrics of different partitionings.

Metric → Partitioning ↓	Optimality gap [%]	Parallel execution time [s]	Number of agents	Core seconds[s]	Cost gap [%]
$\mathcal{P}_{\text{DMPC}}$	0	113	26	2931	748
$\mathcal{P}_{\text{Opt},1}$	3.56	88	10	884	156
$\mathcal{P}_{\text{Opt},2}$	4.13	74	8	594	72
$\mathcal{P}_{\text{Rnd}}$	13.34	103	8	826	139
$\mathcal{P}_{\text{Geo}}$	15.44	43	8	346	0

we define 8 geographical areas by manual inspection on the network topology in **Figure 2**. We call this partitioning $\mathcal{P}_{\text{Geo}}$.

As a final comparison strategy, we select a random network partitioning, still with 8 control agents. We denote this as $\mathcal{P}_{\text{Rnd}}$. For each case, we use a static partitioning approach. This choice is motivated by the fact that the system at study is linear, and therefore the topology of its associated graph does not vary with time.

D. Partitioning Results and Discussion

For each of the five partitionings, we have executed the DMPC-ADMM strategy to control the EEA-ENB for the equivalent of one hour of network data using the same tuning parameters. We report the results of these executions in **Table I**, which we discuss in the following.

The first column is the optimality gap. In this aspect, partitioning $\mathcal{P}_{\text{DMPC}}$ is the best performer, and it is used as a reference. All other partitionings perform worse. However, the results for $\mathcal{P}_{\text{Opt},1}$, and $\mathcal{P}_{\text{Opt},2}$, which are obtained through optimization, are relatively close with a gap of less than 4.13%. For $\mathcal{P}_{\text{Geo}}$ and $\mathcal{P}_{\text{Rnd}}$ we start seeing a degradation of the performance with a gap of more than a 13.34%, which is much worse than the others. The second column is the execution time required to run the algorithm in parallel using the number of cores reported in the third column. In this category $\mathcal{P}_{\text{Opt},1}$ and $\mathcal{P}_{\text{Opt},2}$ have a significantly lower execution time than $\mathcal{P}_{\text{DMPC}}$ with a comparably small loss in performance. The fourth column represents the number of core seconds necessary to implement the strategy, and it is obtained as the product of the number of cores times the parallel execution time. This measure is commonly used in parallel computing to express computation resource cost. We further clarify that core seconds are computed taking at each iteration the slowest performing agent. We sum the computation times of the slowest performer for each step to obtain the parallel execution time, which is the time that a number of cores equal to the number of agents have to operate in parallel to execute the algorithm in the least amount of time. When the algorithm is running, it occupies a number of computation resources (e.g. CPU cores) equal to the number of agents for an amount of time equal to the parallel computation time. With the product of these two factors, we obtain the amount of core seconds necessary to run the algorithm.

The value of the cost gap reported in the last column represents how expensive it is to run the algorithm in parallel in terms of computation resources cost w.r.t. $\mathcal{P}_{\text{Geo}}$, which has the lowest computation resources cost, and we take it as a reference in this category. The strategy $\mathcal{P}_{\text{Geo}}$ is the cheapest in terms of computation resources cost, but it also is the worst in terms of control performance. With strategy $\mathcal{P}_{\text{Opt},2}$ we have an increase of 11.31% in performance from $\mathcal{P}_{\text{Geo}}$, but a computation cost raise of 72%. To further improve the performance of a 0.57% from $\mathcal{P}_{\text{Opt},2}$ to $\mathcal{P}_{\text{Opt},1}$, we need an additional computation cost of 84%. Strategy $\mathcal{P}_{\text{DMPC}}$ is the most expensive, but it also provides the best performance. Lastly, the random partitioning $\mathcal{P}_{\text{Rnd}}$ has both a high computation cost and bad performance.

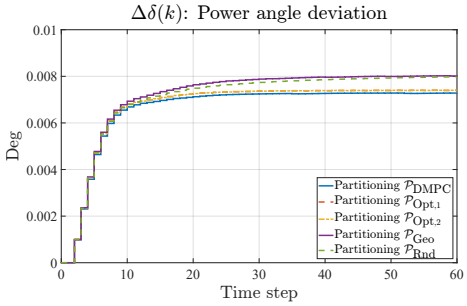

Fig. 3: Evolution of the power angle deviation.

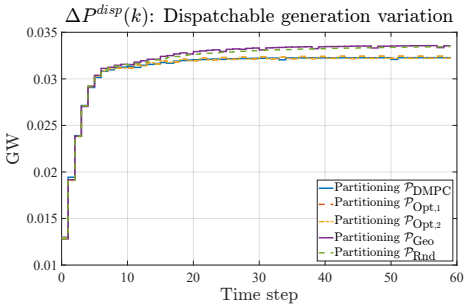

Fig. 4: Evolution of dispatchable generation.

According to these results, we can conclude that the most balanced partitionings are $\mathcal{P}_{\text{Opt},1}$ and $\mathcal{P}_{\text{Opt},2}$. These two strategies provide a comparably small loss in performance, but considerable improvements in computation cost. The selection between $\mathcal{P}_{\text{Opt},1}$ and $\mathcal{P}_{\text{Opt},2}$ can be performed according to the specific needs of the application. Moreover, the cost index (6) can be further refined to suit these needs.

We also present some simulation results for the EEA-ENB controlled with DMPC-ADMM for the different partitionings. The EEA-ENB consists of 26 areas, thus visualizing the state of each area on a single graph for five different partitionings is prohibitive. Therefore, to improve the interpretation of the results, we decided to show the evolution of the norm of each state for each partitioning. In **Figure 3** we report the evolution of the power angle deviation from the nominal value. The results reflect the situation reported in **Table I**, with $\mathcal{P}_{\text{DMPC}}$ providing the smallest angle deviation, and $\mathcal{P}_{\text{Opt},1}$, $\mathcal{P}_{\text{Opt},2}$ closely following. Also, the gap with $\mathcal{P}_{\text{Rnd}}$ and $\mathcal{P}_{\text{Geo}}$ is evident in this regard. This situation is evident also in **Figure 4**, where we report the variation in dispatchable power allocation required to compensate for the load demand.

XI. CONCLUSIONS AND FUTURE WORK

In this article, we have introduced a general method for partitioning and we have shown how an accurate selection of the partitioning in the application of distributed control can significantly reduce the computation speed and resource cost, with marginal performance degradation. To do this, first, we introduced the basic definition of *atomic control agent* and *control agent* to construct a generalized partitioning strategy, and we defined an algorithm to select the atomic control agents, and a global specialized partitioning metric. Then, this metric is used to partition the system through an integer program that returns both the number of sets and the elements in the sets of the partitioning. We provided a case study for the distributed predictive control of power systems in which we tested different partitionings, obtaining with our novel strategy, considerable improvements in the computation resources required to solve the distributed problem. Future work could extend the mathematical framework for the generalized partitioning technique with formal definitions of *atomic control agent* and *control agent*, extend it beyond linear systems, and perform an extensive assessment of the proposed method for a wide range of distributed control benchmarks. Providing a closed form for the optimization metric is another direction of interest. Also, ensuring control properties, such as controllability of the agents resulting from partitioning remains an open issue. Moreover, in a more general setting where online re-partitioning of large-scale networks is required, it is important to develop efficient online distributed optimization strategies for the partitioning problem.

ACKNOWLEDGMENTS

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

REFERENCES

[1] D. D. Šiljak, *Decentralized control of complex systems*. Academic Press, INC., 1991.

[2] R. Scattolini, "Architectures for distributed and hierarchical model predictive control – a review," *Journal of Process Control*, vol. 19, pp. 723–731, 2009.

[3] P. Chanfreut, J. M. Maestre, and E. F. Camacho, "A survey on clustering methods for distributed and networked control systems," *Annual Reviews in Control*, vol. 52, pp. 75–90, 2021.

[4] R. Rostami, G. Costantini, and D. Görges, "Admm-based distributed model predictive control: Primal and dual approaches," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 6598–6603.

[5] A. Riccardi, L. Laurenti, and B. De Schutter, "A benchmark for the application of distributed control techniques to the electricity network of the european economic area," in *Control Systems Benchmarks: Key Resources for Researchers and Practitioners*. Springer, 2024, dOI: https://doi.org/10.48550/arXiv.2403.14372.

[6] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, pp. 75–174, 2010.

[7] P. Segovia, V. Puig, E. Duviella, and L. Etienne, "Distributed model predictive control using optimality condition decomposition and community detection," *Journal of Process Control*, vol. 99, pp. 54–68, 2021.

[8] W. Tang, A. Allman, D. B. Pourkargar, and P. Daoutidis, "Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection," *Computers & Chemical Engineering*, vol. 111, pp. 43–54, 2018.

[9] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Physics Reports*, vol. 533, pp. 95–142, 2013.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, p. P10008, 2008.

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

[12] J. Barreiro-Gomez, *Partitioning for large-scale systems: Sequential DMPC design*. Springer, 2019, pp. 163–178.

[13] F. Fele, J. M. Maestre, and E. F. Camacho, "Coalitional control: Cooperative game theory and control," *IEEE Control Systems Magazine*, vol. 37, pp. 53–69, 2017.

[14] F. Fele, E. Debada, J. M. Maestre, and E. F. Camacho, "Coalitional control for self-organizing agents," *IEEE Transactions on Automatic Control*, vol. 63, pp. 2883–2897, 2018.

[15] P. Chanfreut, J. M. Maestre, T. Hatanaka, and E. F. Camacho, "Fast clustering for multi-agent model predictive control," *IEEE Transactions on Control of Network Systems*, vol. 9, pp. 1544–1555, 2022.

[16] W. Ananduta, T. Pippia, C. Ocampo-Martinez, J. Sijs, and B. De Schutter, "Online partitioning method for decentralized control of linear switching large-scale systems," *Journal of the Franklin Institute*, vol. 356, pp. 3290–3313, 2019.

[17] W. Ananduta and C. Ocampo-Martinez, "Event-triggered partitioning for non-centralized predictive-control-based economic dispatch of interconnected microgrids," *Automatica*, vol. 132, p. 8, 2021.

[18] P. R. Baldivieso Monasterios, P. A. Trodden, and M. Cannon, "On feasible sets for coalitional mpc," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 4668–4673.

[19] A. Riccardi, L. Laurenti, and B. De Schutter. (2024) Code underlying the publication: A generalized partitioning strategy for distributed control. DOI: https://doi.org/10.4121/90ada13d-a6c9-4e4c-a046-2b984595bcdd.

[20] ——. (2023) Code underlying the publication: A benchmark for the application of distributed control techniques to the electricity network of the european economic area. DOI: https://doi.org/10.4121/d2c0d075-1c49-41af-8113-5e50c27ca97e.

[21] Entso-e. The European network of transmission system operators.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, p. 122, 2010.

[23] T. H. Summers and J. Lygeros, "Distributed model predictive consensus via the alternating direction method of multipliers," in *50th Annual Allerton Conference on Communication, Control, and Computing*, 2012, pp. 79–84.