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

Resilience of Time-Varying Communication Graphs for Consensus of Changing Sets of Computing Agents

Vincent Schmidtke[1] Zonglin Liu[1] and Olaf Stursberg[1]

***Abstract*— System performance of distributed control systems and networked computing systems is strongly dependent of the underlying communication topology. This paper considers the rarely studied problem of how the topology can maintain resilience by reconfiguration in case that agents leave or join the network during online operation. Existing optimization-based approaches which reconfigure the entire network can typically not be used in this case, since the computational burden for online application is too high. Thus, this paper proposes a novel combined offline-online scheme which optimizes the topology for high convergence rate (of e.g. consensus problems) while providing guarantees for the robustness against agent failures. In the offline part, an optimization of the entire topology is carried out using novel constraints to prepare resilience of the online procedure. For the latter, the proposed scheme guarantees that robustness is maintained for joining agents and if a specified number of agents leave the network. In simulation, the proposed scheme is compared to existing approaches and the advantages of the online-offline procedure are demonstrated.**

I. Introduction

Multi-agent systems can be found in various applications today, most of which are large-scale systems, such as sets of autonomous vehicles, UAV swarms, or autonomous industrial robots. The individual agents are often assumed to have their own computing unit and can exchange information over a communication network. Among others, consensus problems for a group of networked computing agents are often considered in literature [1]–[3]: Let a set $\mathcal{N} = \{1, \dots, N\}$ of agents be given which communicate over an undirected graph $G = (\mathcal{N}, \mathcal{E})$, with $\mathcal{E}$ containing the available communication edges. For a connected graph G and arbitrary initialization of the agents' states $x_i(t_0) \in \mathbb{R}^n$, $i \in \mathcal{N}$ at time t_0, consider the goal of reaching consensus, i.e., the local variables $x_i(t)$ converge to a common value $\frac{1}{N}\sum_{j=1}^{N} x_i(t_0)$ for $t \to \infty$ when using the simple update rule:

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} (x_j(t) - x_i(t)),\ t \geq t_0, \tag{1}$$

in which $\mathcal{N}_i \subseteq \mathcal{N}$ denotes the set of agents that can communicate with i.

Now focus on the case that the set of agents participating in the task varies over time, as may occur, e.g., due to malfunction of certain agents, or if new agents try to join the set. Since the consensus value $\frac{1}{N}\sum_{j=1}^{N} x_i(t_0)$ depends explicitly on the participating agents, a changing set of agents first of all alters the value. Even if this is acceptable (since a consensus is still reached), the following problems may occur when using (1): 1). The graph may loose connectivity if certain agents leave the network. 2). Given a current graph, it may not be possible to establish links to joining agents due to limitations of the communication bandwidth (possibly instantiated by a maximum number of connections per agent). 3). A reconfiguration of the existing graph may be costly and consume too much time.

Although the topic of resilient consensus for time-varying agents is important for applications such as vehicle coordination or plug-and-play scenarios in electrical networks, this problem has found only little attention in research so far. For most work considering consensus problems with time-varying graph, the set of participating agents is fixed, while the communication edges are optimized in order to increase the convergence rate, see [4], [5], [6]. The optimization of the edges in that work is achieved by solving mixed-integer semi-definite programming (SDP) problems, which theoretically can be applied to reconfigure existing graphs for the given problem. But high computational complexity often prevents the application, in particular for large sets of agents, and if the graph has to be reconfigured fastly. This situation was confirmed in [7] for a large group of drones, where the graph has to be reconfigured upon failure of single drones in order to preserve connectivity. Efficient heuristics developed in the last years (see [8] and [9]) may be applied to reduce the computation time, but frequent reconfiguration of the graph also poses challenges to these approaches. In two recent papers [10] and [11], methods to reconfigure the graph for joining or leaving agents were introduced; in [11] no optimization problem has to be solved for reconfiguration, but the graph has to fulfil restrictive assumptions (such as even numbers $|\mathcal{N}_i|$).

To counteract negative consequences caused by changes of the set of computing agents, the proposal in this paper is to first optimize an initial graph by solving an mixed-integer semi-definite programming (SDP) problem. This aims at ensuring that the initial graph is robust against loss of connectivity for the case of leaving agents. If agents are joining in the online process, a strategy to quickly determine new edges connecting to the existing graph is proposed, while no reconfiguration has to be carried out. It is shown that the resulting graph can always maintain the same robustness as the initial graph, even if arbitrarily many new agents join the network.

In the next section, the problem of time-varying agents is described in detail, while the considered offline and online strategies are introduced in Sec. 3 and 4. Effectiveness and

*This work has been financially supported in part by the German Research Foundation (DFG) through the project *CoInCiDE*.

[1]Control and System Theory, EECS, University of Kassel, Germany {v.schmidtke, z.Liu, stursberg}@uni-kassel.de

efficiency of the proposed method comparing to existing ones are demonstrated by numerical examples in Sec. 5, and the paper concludes in Sec. 6.

II. Graphs for Time-Varying Sets of Agents

Given an undirected communication graph G of a considered set of agents, the adjacency matrix $A(G) \in \mathbb{R}^{N\times N}$ of G is determined by setting the entry a_{ij} in the i-th row and j-th column of $A(G)$ equal to one if an edge e_{ij} exists, and zero otherwise. Based on $A(G)$, the Laplacian matrix of the graph is determined by $L(G) := D(G) - A(G)$, where $D(G) \in \mathbb{R}^{N\times N}$ is a diagonal matrix with $d_{ii} = \sum_{j=1}^{N} a_{ij}$ on the diagonal. After collecting all local consensus variables $x_i(t)$, $i \in \mathcal{N}$ into a global vector $x(t) := [x_1^T(t), x_2^T(t), \ldots, x_N^T(t)]^T \in \mathbb{R}^{N\cdot n\times 1}$, the dynamics of $x(t)$ is defined to:

$$\dot{x}(t) = -L(G)x(t), \quad t \geq t_0, \tag{2}$$

which depends explicitly upon the Laplacian matrix. The eigenvalues of $L(G)$ satisfy:

$$0 = \lambda_1(L(G)) \leq \lambda_2(L(G)) \leq \ldots \leq \lambda_n(L(G)) \tag{3}$$

and $\lambda_2(L(G)) > 0$ applies if the graph G is connected [12]. In addition, $\lambda_2(L(G))$ also provides an upper-bound of the convergence rate in (2).

For most consensus problems in practice, the (heterogeneous) communication bandwidth of each agent is also limited [13], implying that only a few communication edges can be maintained at any time. To this end, the following assumption is introduced in this work:

Assumption 1: For any agent $i \in \mathcal{N}$, the number of admissible (undirected) communication edges is limited to $n_i \in \mathbb{N}^{\geq 1}$, and $|\mathcal{N}_i| \leq n_i$ always holds.

In case agents are leaving or joining the consensus process, a time-varying graph $G_{t_k} = (\mathcal{N}_{t_k}, \mathcal{E}_{t_k})$ is employed, where $t_k > t_0$ $k \in \{1, 2, \ldots\}$ records the time for each change of the set of agents. Note that the situation that more than one agent leaves or joins at the same time can be regarded as a sequence of single changes without progress of time.

If the update rule (1) is applied to the case of time-varying graphs, the following problems may occur (in addition to variations of consensus value and convergence rate):

1) The leaving of agents may disconnect a previously connected graph (see Fig. 1), such that a consensus cannot be reached any longer through (1);
2) The joining of new agents is not possible, if no additional communication edges can be added to the existing graph due to the restriction formulated in Assumption 1, see Fig. 2;
3) To address the two problems above, the edges in the current graph must be reconfigured, what can be very complicated for large numbers of agents (especially if a certain convergence rate must be maintained).

To address these problems, the edges in the initial graph G_{t_0} are first optimized offline such that the graph remains connected for a bounded number of agents leaving, and has sufficient capacity for additional edges to accommodate new agents. In the online phase, only the edges for connecting new agents have to be optimized (a task of relatively low complexity), while the properties of the initial graph are recursively guaranteed for arbitrarily many joining agents.

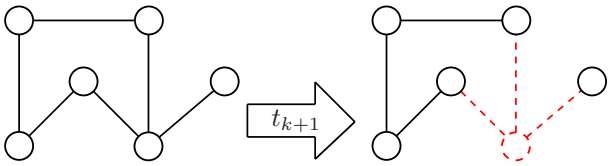


Fig. 1: From t_k to t_{k+1}, the graph has lost its connectivity due to the leaving agent marked by red dashed lines.

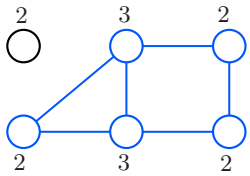


Fig. 2: The blue graph G_{t_k} cannot connect to the joining agent shown as a black circle in t_{k+1} without further reconfiguration, since all agents have reached their maximum number n_i of communication edges (listed next to the corresponding nodes).

III. Offline Optimization of the Initial Graph G_{t_0}

In the offline design phase, the communication edges for the initial group of $N_{t_0} := |\mathcal{N}_{t_0}|$ agents are optimized by solving a mixed-integer SDP problem. In detail, a binary variable $b_{ij} \in \{0, 1\}$ is used to encode whether an agent i can communicate with agent j with $i, j \in \mathcal{N}_{t_0}$. For $b_{ij} = 1$, an edge exists between the two agents, and $b_{ij} = 0$ otherwise. In addition, as the communication is undirected, $b_{ij} = b_{ji}$ holds for all $i, j \in \mathcal{N}_{t_0}$. By using the binary variables, the adjacency matrix $A(G_{t_0})$ of the graph can be encoded by:

$$A(G_{t_0}) := \begin{bmatrix} 0 & b_{12} & \cdots & b_{1N_{t_0}} \\ b_{21} & 0 & \ddots & b_{2N_{t_0}} \\ \vdots & \ddots & \ddots & \vdots \\ b_{N_{t_0}1} & b_{N_{t_0}2} & \cdots & 0 \end{bmatrix}, \tag{4}$$

and the Laplacian matrix is:

$$L(G_{t_0}) := \begin{bmatrix} \sum\limits_{j\in\mathcal{N}_{t_0}} b_{1j} & -b_{12} & \cdots & -b_{1N_{t_0}} \\ -b_{21} & \sum\limits_{j\in\mathcal{N}_{t_0}} b_{2j} & \ddots & -b_{2N_{t_0}} \\ \vdots & \ddots & \ddots & \vdots \\ -b_{N_{t_0}1} & -b_{N_{t_0}2} & \cdots & \sum\limits_{j\in\mathcal{N}_{t_0}} b_{N_{t_0}j} \end{bmatrix}. \tag{5}$$

Given an integer-valued constant $\rho > 1$, a sufficiently large constant $m > 0$ (that is larger than the left-hand side of (11)), and a matrix $\mathbf{1}\mathbf{1}^T$ of ones with dimension $N_{t_0} \times N_{t_0}$, the

following mixed-integer SDP problem is introduced, in order to determine the communication edges in G_{t_0}:

Problem 1:

$$\max_{\{b_{ij}\},\{c_{il}\},\gamma} \gamma \tag{6}$$

$$\text{s.t.: } \gamma \cdot I \preceq L(G_{t_0}) + \gamma \cdot \mathbf{1}\mathbf{1}^T,\ \gamma \in \mathbb{R} \tag{7}$$

$$b_{ij} \in \{0,1\},\ b_{ij} = b_{ji},\ \forall i,j \in \mathcal{N}_{t_0} \tag{8}$$

$$\sum_{j \in \mathcal{N}_{t_0}} b_{ij} \leq n_i,\ \forall i \in \mathcal{N}_{t_0} \tag{9}$$

$$\gamma > \rho - 1 \tag{10}$$

$$n_i - \sum_{j \in \mathcal{N}_{t_0}} b_{ij} \geq l + m(c_{il} - 1), \forall l \in \{1,\dots,\rho\}, \forall i \in \mathcal{N}_{t_0} \tag{11}$$

$$\sum_{i \in \mathcal{N}_{t_0}} c_{il} = 1,\ \ \forall l \in \{1,\dots,\rho\} \tag{12}$$

$$\sum_{l=1}^{\rho} c_{il} \leq 1,\ \forall i \in \mathcal{N}_{t_0}. \tag{13}$$

Note that without the constraints (10) – (13), Problem 1 would be a standard problem for maximizing the second largest eigenvalue $\lambda_2(L(G_{t_0}))$ of the Laplacian matrix: As the eigenvalues of the (symmetric) matrix $\gamma \cdot \mathbf{1}\mathbf{1}^T$ are $\{0, 0, \dots, 0, \gamma\}$, the matrix inequality (7) ensures that γ provides a lower bound of $\lambda_2(L(G_{t_0}))$ according to the Weyl theorem [14]. By maximizing γ in (6), the eigenvalue $\lambda_2(L(G_{t_0}))$ is thus also maximized, leading to a faster convergence of (2) in case no agent is joining and leaving the initial graph G_{t_0}.

The constraint (10) in Problem 1 defines a lower bound of the optimized γ. This constraint aims at ensuring the connectivity of the graph when agents are leaving the initial graph. The notion of *vertex connectivity* of a graph is introduced according to [15]:

Def. 1 (Vertex Connectivity): For an undirected and connected graph $G = (\mathcal{N}, \mathcal{E})$, the vertex connectivity is defined to $\kappa(G) := \rho \in \mathbb{N}^{\geq 1}$, if at least ρ agents must be removed from $\mathcal{N}$ in order to disconnect the graph.

Based on this definition, the following fact is established for the initial graph G_{t_0}:

Theorem 1: For any graph G_{t_0} satisfying the constraints (7) – (10) in Problem 1, the corresponding vertex connectivity satisfies $\kappa(G_{t_0}) \geq \rho$.

Proof. According to [12], it is known that if the undirected graph G_{t_0} is connected, the relation:

$$\lambda_2(L(G_{t_0})) \leq \kappa(G_{t_0}) \tag{14}$$

between the Laplacian matrix $L(G_{t_0})$ and the vertex connectivity $\kappa(G_{t_0})$ applies. As $\gamma \leq \lambda_2(L(G_{t_0}))$ applies according to (7), as well as $\rho - 1 < \gamma$ according to (10), the following inequalities must hold for G_{t_0}:

$$\rho - 1 < \gamma \leq \lambda_2(L(G_{t_0})) \leq \kappa(G_{t_0}). \tag{15}$$

As the vertex connectivity $\kappa(G_{t_0})$ is integer-valued according to Def. 1, the relation $\kappa(G_{t_0}) \geq \rho$ must apply according to (15). □

The constraint (10) thus ensures that even if $\rho - 1$ arbitrary agents leave the initial graph G_{t_0} the remaining graph is still connected.

Remark 1: The vertex connectivity is a lower bound for the edge connectivity [12]. The latter can be defined equivalently to $\kappa(G)$ in Def. 1, but referring to the number of edges to be removed to loose connectvity. The whole procedure is therefore guaranteed to provide robustness with respect to failures of communication edges as well.

With regard to the constraints (11) to (13) in Problem 1, a set of additional binary variables $c_{il} \in \{0,1\},\ \forall l \in \{1,\dots,\rho\},\ \forall i \in \mathcal{N}_{t_0}$ is introduced. These constraints serve the purpose of ensuring that a number of ρ different agents exist, such that their local capacity of free edges $\xi_l := n_l - \sum_{j \in \mathcal{N}_{t_0}} b_{lj,t_0}, l \in \{1,\dots,\rho\}$ in G_{t_0} satisfies:

$$\xi_l \geq l,\ \ \ l \in \{1,\dots,\rho\}. \tag{16}$$

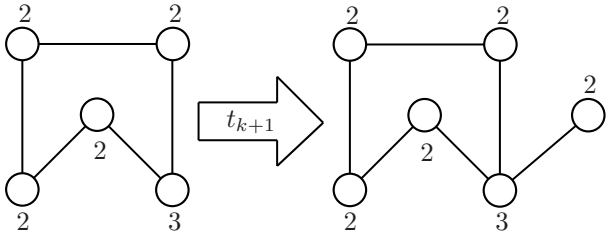
As it is unknown in advance which agent from $\mathcal{N}_{t_0}$ needs to satisfy (16), the latter constraint is encoded by binary variables c_{il} in (11) – (13): For each agent $i \in \mathcal{N}_{t_0}$, a number of ρ binary variables $c_{il} \in \{0,1\},\ \forall l \in \{1,\dots,\rho\}$ are assigned, in order to identify if i belongs to one of the ρ agents satisfying (16). The value $c_{il} = 0,\ \forall l \in \{1,\dots,\rho\}$ implies that the agent i does not belong to any of the ρ agents. The constraint (13) ensures that the agent i can only be assigned to one of the ρ agents satisfying (16), while the constraint (12) guarantees that for any $l \in \{1,\dots,\rho\}$, the condition $\xi_l \geq l$ in (16) is satisfied by at least one agent in $\mathcal{N}_{t_0}$. The constraint (11) adopts the so called *Big-M* relaxation method [16] in order to ensure that the constraint (16) is either satisfied or relaxed for each agent in $\mathcal{N}_{t_0}$.

Note that if the initial graph G_{t_0} satisfies (16), a new joining agent can be connected with G_{t_0} by constructing an edge with any of the ρ agents. Furthermore, it will be shown in the next section, that by a suitable selection of the edges connecting with the new agent, the vertex connectivity of the resulting graph can also be maintained in addition (which is critical if the agents are leaving thereafter).

Remark 2: Communication costs can be straightforwardly included in the cost function in Problem 1. To do so, a scalar cost $\beta_{ij} \in \mathbb{R}^+$ is assigned to each binary variable b_{ij}. The cost values would be motivated by the underlying problem and could encode, e.g., distances between agents. The constraints (10) – (13) in addition ensure that the online process (to be described next) is prepared with respect to robustness and sufficient free capacity for additional edges.

IV. Online Part for Changing Sets of Agents

Agents can join or leave the initial graph G_{t_0} in the online process. Two cases are considered, where agents can only join the existing graph G_{t_k} in the first case. In the second case agents can join and leave the current graph G_{t_k}, assuming that only a single agent is joining or leaving at any time instance.



$\kappa(G_{t_k}) = 2$ $\kappa(G_{t_{k+1}}) = 1$

Fig. 3: Due to a low capacity of free edges in G_{t_k}, the new agent can only join the graph in the shown manner, leading to a decrease of vertex connectivity of the graph in t_{k+1}. (n_i is stated next to each agent.)

A. Joining Agents

An online optimization method is proposed in the following for the case that agents only join the network. Since using a complete reconfiguration of the graph (as in the previous section) is time-consuming, a simpler local change of the initial graph is chosen. Two critical situations may occur (and are addressed in the proposed solution):

1.) the present graph does not provide any free edge to connect to the new agent;
2.) the joining of new agents may significantly worsen the vertex connectivity of the graph, and thus makes the latter fragile to the leaving of agents, see Fig. 3.

Let the initial graph $G_{t_0} = (\mathcal{N}_{t_0}, \mathcal{E}_{t_0})$ be given and assume that a new agent s with $n_s \geq 2\rho$ is joining at time t_1. By using a binary variable $b_{sj}(= b_{js}) \in \{0, 1\}$ to denote the existence of a communication edge between s and any $j \in \mathcal{N}_{t_0}$, the Laplacian matrix of the graph G_{t_1} is given by:

$$L(G_{t_1}) = \begin{bmatrix} L(G_{t_0}) + diag(b_{1s}, \ldots, b_{N_{t_0}s}) & \begin{bmatrix} -b_{1s} \\ \vdots \\ -b_{N_{t_0}s} \end{bmatrix} \\ \begin{bmatrix} -b_{s1} & \cdots & -b_{sN_{t_0}} \end{bmatrix} & \sum\limits_{j \in \mathcal{N}_{t_1}} b_{sj} \end{bmatrix}. \tag{17}$$

Based on $L(G_{t_1})$, the following optimization is solved to determine the edges connecting with the agent s:

Problem 2:

$$\max_{\{b_{sj}\}, \gamma} \ \gamma \tag{18}$$

$$\text{s.t.: } \gamma \cdot I \preceq L(G_{t_1}) + \gamma \cdot \mathbf{1}\mathbf{1}^T, \ \gamma \in \mathbb{R}, \tag{19}$$

$$b_{sj}, b_{js} \in \{0, 1\}, b_{sj} = b_{js}, \ \forall j \in \mathcal{N}_{t_0} \tag{20}$$

$$\sum_{j \in \mathcal{N}_{t_0}} b_{sj} \leq n_s \tag{21}$$

$$\sum_{j \in \mathcal{N}_{t_0}} b_{sj} \geq \rho \tag{22}$$

$$\xi_l \geq l, \ l \in \{1, \ldots, \rho\}, \ \text{for } \rho \text{ different agents in } \mathcal{N}_{t_1}. \tag{23}$$

Note that without the constraints (22) and (23)[1], the solution of Problem 2 once more aims at maximizing $\lambda_2(L(G_{t_1}))$ and thus the convergence rate of (2). Considering the constraints (22) and (23) in addition, leads to the following theorem:

Theorem 2: Let the initial graph G_{t_0} obtained from the solution of Problem 1 be given. If $n_s \geq 2\rho$ applies for the new agent s, then Problem 2 has a feasible solution, and the vertex connectivity of the graph G_{t_1} resulting from this solution satisfies $\kappa(G_{t_1}) \geq \rho$.

Proof. It is known from Problem 1 that a group of ρ agents in $\mathcal{N}_{t_0}$ exists for which the capacity of free edges satisfies: $\xi_l \geq l, l \in \{1, \ldots, \rho\}$. By connecting the new agent s to each agent in this group, the free edge capacity of the latter then satisfies: $\xi_l \geq l-1, l \in \{1, \ldots, \rho\}$. In addition, since $n_s \geq 2\rho$ applies for the new agent s, the capacity $\xi_s \geq \rho$ must hold after connecting with the ρ agents in $\mathcal{N}_{t_0}$. As a result, a new group of ρ agents in $\mathcal{N}_{t_1}$ (including the new agent s) is found, which satisfy the constraint (23) in Problem 2., i.e. a candidate for a feasible solution of the latter problem is found.

To show the vertex connectivity of G_{t_1}, recall from Theorem 1 that by removing arbitrary $\rho-1$ agents from G_{t_0}, the remaining graph is still connected. Thus, by removing arbitrary $\rho - 1$ agents from G_{t_1}, and in case all removed agents are contained in the previous graph G_{t_0}, the remaining agents in G_{t_0} together with the new agent s must be further connected. This holds since the constraint (22) enforces that the agent s connects with at least ρ agents in G_{t_0}. For the different case in which the removed $\rho - 1$ agents contain the new agent s, this is equivalent to removing $\rho - 2$ agents from G_{t_0}. As a result, the remaining agents must also stay connected. Accordingly, the vertex connectivity $\kappa(G_{t_1}) \geq \rho$ holds for the graph G_{t_1} from Problem 2. □

Note that Theorem 2 also applies for the case that more than one agent enter the graph:

Corollary 1: Given the initial graph G_{t_0} obtained from Problem 1, if the newly joining agent s in any following time step t_k, $k \in \{1, 2, \ldots\}$ satisfies $n_s \geq 2\rho$, then the Problem 2 (with G_{t_0} being replaced by $G_{t_{k-1}}$) remains feasible, and $\kappa(G_{t_k}) \geq \rho$ applies for the resulting graph G_{t_k}.

Corollary 1 establishes a recursive guarantee for agents joining into the graph, i.e., the time-varying graph always has the capacity to accommodating new agents, while also being robustly connected when agents are leaving.

B. Complexity of the Optimization Problems

The use of binary variables in Problem 1 and 2 makes both of them NP-hard problems. Specifically, a number of $|\mathcal{N}_{t_0}| \times (\rho + |\mathcal{N}_{t_0}|)$ binary variables are used in Problem 1, while reduced to $|\mathcal{N}_{t_0}| \times (\rho + 2)$ in Problem 2. As Problem 1 is solved in the offline phase, a longer computation time can be regarded as acceptable and many heuristic or distributed solution strategies developed in the last years can be applied to accelerate the solution process (see for instance [8] or [9]).

For Problem 2, although only the edges connecting with the new agent are optimized, the problem may still require significant time for the solution process. In the worst case, the new agent has to wait until Problem 2 is solved, before

[1]The constraint (23) can be similarly encoded by additional binary variables as in Problem 1 and they are not shown here for brevity.

it can participate into the consensus process. Regarding this problem, the feasible solution candidate of Problem 2, which is described in the proof to Theorem 2, can be adopted to reduce the waiting time. This is due to the fact that a feasible solution only differs in the convergence rate comparing to the optimal one, while all desired properties of the resulting graph in Corollary 1 are preserved. After the optimum of Problem 2 has been found, one can switch to the optimal edges in order to attain faster convergence.

C. Joining and Leaving Agents

For the initial graph satisfying $\kappa(G_{t_0}) \geq \rho$ from Problem 1, it is known that $\kappa(G_{t_k}) \geq \rho - N_l$, $N_l < \rho$, holds if N_l agents have left in the time interval $[t_0, t_k]$. Assume now that a new agent intends to join at time t_{k+1}.

Lemma 1: For G_{t_k} with $\kappa(G_{t_k}) \geq \rho - N_l$, if the joining agent satisfies $n_s \geq 2(\rho - N_l)$, then the problem:

Problem 3:

$$\max_{\{b_{sj}\},\gamma} \ \gamma \tag{24}$$

$$\text{s.t.: } \gamma \cdot I \preceq L(G_{t_{k+1}}) + \gamma \cdot \mathbf{1}\mathbf{1}^T, \ \gamma \in \mathbb{R}, \tag{25}$$

$$b_{sj}, b_{js} \in \{0,1\}, b_{sj} = b_{js}, \ \forall j \in \mathcal{N}_{t_k} \tag{26}$$

$$\sum_{j \in \mathcal{N}_{t_k}} b_{sj} \leq n_s \tag{27}$$

$$\sum_{j \in \mathcal{N}_{t_k}} b_{sj} \geq \rho - N_l \tag{28}$$

$$\xi_l \geq l, l \in \{1, \ldots, \rho - N_l\}, \text{for } \rho - N_l \text{ agents in } \mathcal{N}_{t_{k+1}} \tag{29}$$

has a feasible solution, and the resulting graph $G_{t_{k+1}}$ satisfies $\kappa(G_{k+1}) \geq \rho - N_l$.

For the free capacity in G_{t_k}, (29) holds for $\rho - N_l$ agents in $\mathcal{N}_{t_k}$. To show this consider first that none of the leaving agents are part of (16), implying directly (29) for $\mathcal{N}_{t_k}$. Now consider that all N_l agents belong to the set satisfying (16). Since the maximal value of l in G_{t_k} is $\rho - N_l$ and $\xi_i > \xi_j$ for $i > j$, (29) holds for $\rho - N_l$ agents in $\mathcal{N}_{t_k}$. Based on these two extreme cases all other possibilities of leaving agents can be shown trivially. With this property for G_{t_k} together with $n_s \geq 2(\rho - N_l)$, the vertex connectivity and the guarantee of a feasible solution for Problem 3 can be shown analogously to the proof of Theorem 2 and Corollary 1. Lemma 1 ensures that worst-case robustness (i.e., $\kappa(G_{t_k}) - \rho = 1$) is always preserved if some agents leave the graph.

If the worst-case robustness gets 1, Problem 1 can be used to reconfigure the graph.

V. Numerical Example

In this section, the effectiveness of the proposed optimization problems is shown for a numeric example. For online optimization, the proposed optimization problem is compared with the 'Best Algebraic' (BA) approach by [10], which results from Problem 2 by omitting the constraints (22) and (23).

In the offline optimization $N_{t_0} = 7$ agents are considered.

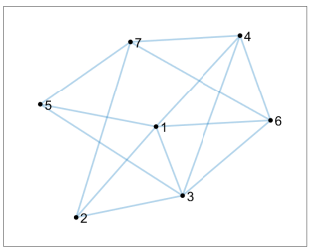


Fig. 4: Resulting Graph G_{t_0} from Problem 1.

The maximal number of edges for every agent $i \in \mathcal{N}_{t_0}$ is chosen as $n_i = 6$, and the parameters $\rho = 3$ and $m = 20$ are used. The resulting graph is shown in Fig. 4 and has a second smallest eigenvalue $\lambda_2(G_{t_0}) = 3$, and a vertex connectivity of $\kappa(G_{t_0}) = 3$. Solving Problem 1 with YALMIP [17] took 35.79s using a processor with 16-GB RAM and a 4-GHz Intel core i7-4790K.

Considering the online process, Theorem 2 and Corollary 1 apply for $n_s \geq 6$. Solving Problem 2 ten times in a row with $n_s = 7$ leads to the graph in Fig. 5. The graph resulting from the BA approach is shown in Fig. 6 for comparison. Both the second smallest eigenvalue and the vertex connectivity are plotted in Fig. 7 over the number of joined agents for the two approaches. BA is a greedy approach where as many edges as possible are established in each t_k to maximize the eigenvalue of the resulting graph, leading to larger values for both properties for the first joining agents. The capacity of free edges in the graph decreases after the first agents join the graph, leading to the case where agents can only establish few edges to the existing graph, For example, agent 11 in Fig. 6 establishes two edges to G_{t_3}, resulting in an immense decrease of the second smallest eigenvalue (see $\lambda_2(G_{t_4})$ in Fig. 7a). Therefore, vertex connectivity is also reducing to two, as deleting the agents 10 and 8 disconnects the graph, see Fig. 6.

The graph obtained from solving Problem 2, however, always satisfies $\kappa(G_{t_k}) = 3$ and performs better in the long run in terms of $\lambda_2(G_{t_k})$ (see Fig. 7). The computation times for Problem 2 range here from 21ms to 421ms, showing the low configuration effort needed for the online process. The results for $n_s = 6$ for both methods are shown in Fig. 8. In this case the BA method is infeasible after three agents have joined the graph, since the graph has no free capacity anymore (G_{t_3} is shown in Fig. 8b). In contrast, the solution from Problem 2 in Fig. 8a stays feasible and guarantees a vertex connectivity $\kappa(G_{t_k}) \geq 3$, since the condition $n_s \geq 2\rho$ from Corollary 1 holds.

VI. Conclusion

This paper proposes a novel offline-online scheme to provide guarantees for graphs with time-varying agents and communication constraints. In the offline phase, the topology is optimized in terms of its second smallest eigenvalue of the Laplacian matrix, which provides guarantees for robustness

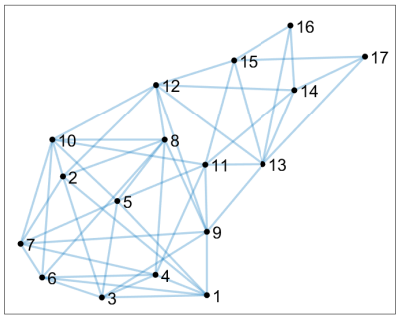


Fig. 5: Resulting graph $G_{t_{10}}$ from the proposed method after 10 agents joined the graph with $n_s = 7$.

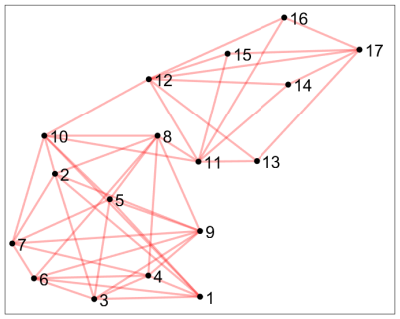
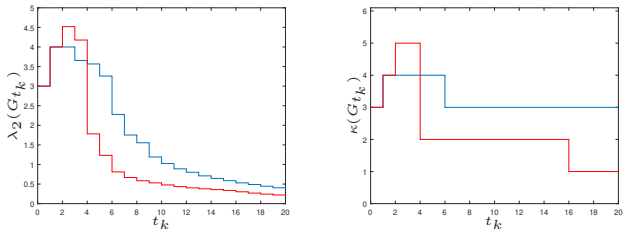


Fig. 6: Resulting graph $G_{t_{10}}$ from the BA method from [10] after 10 agents joined the graph with $n_s = 7$.

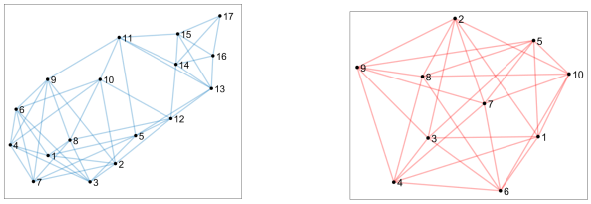


(a) $\lambda_2(G_{t_k})$ (b) $\kappa(G_{t_k})$

Fig. 7: Comparison of the graphs in Fig. 5 (blue) and Fig. 6 (red) with up to 20 joining agents.

against agent failures for the resulting graphs. In the online phase, agents can join and/or leave the graph. First the case of only joining agents, the proposed scheme provides a guarantee for the fact that the vertex connectivity of the graph and the feasibility of problem solution is preserved if the joining agents fulfill certain conditions. These results are extended to the general case of joining and leaving agents using a modified version of optimization problem to be solved online, and it discussed and shown for examples that the computational effort for the online part is moderate.

For future work, the extension to directed graphs as well as techniques for edge rewiring to increase the value of $\lambda_2(G_{t_k})$ are matter of research.



(a) From Problem 2. (b) BA Method from [10].

Fig. 8: Comparison of both methods for G_{t_0} from Fig. 4 and with $n_s = 6$ for all joining agents. The BA method gets infeasible for t_4.

References

[1] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems Magazine*, vol. 27, no. 2, pp. 71–82, 2007.

[2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[3] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-UAV system," in *2014 American Control Conf.*, pp. 2077–2082, IEEE, 2014.

[4] M. Rafiee and A. M. Bayen, "Optimal network topology design in multi-agent systems for efficient average consensus," in *IEEE Conf. on Decision and Control*, pp. 3877–3883, 2010.

[5] D. Groß and O. Stursberg, "Optimized distributed control and network topology design for interconnected systems," in *IEEE Conf. on Decision and Control and European Control Conf.*, pp. 8112–8117, 2011.

[6] R. Dai and M. Mesbahi, "Optimal topology design for dynamic networks," in *IEEE Conf. on Decision and Control and European Control Conf.*, pp. 1280–1285, 2011.

[7] R. K. Ramachandran, N. Fronda, and G. S. Sukhatme, "Resilience in multirobot multitarget tracking with unknown number of targets through reconfiguration," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 609–620, 2021.

[8] A. Ghosh and S. Boyd, "Growing well-connected graphs," in *IEEE Conf. on Decision and Control*, pp. 6605–6611, 2006.

[9] T. Anderson, C.-Y. Chang, and S. Martinez, "Maximizing algebraic connectivity of constrained graphs in adversarial environments," in *European Control Conf.*, pp. 125–130, IEEE, 2018.

[10] S. Stüdli, Y. Yan, M. M. Seron, and R. H. Middleton, "Plug-and-play networks: Adding vertices and connections to preserve algebraic connectivity," in *IEEE Conf. on Decision and Control*, pp. 4823–4828, 2021.

[11] S. Stüdli, Y. Yan, M. M. Seron, and R. H. Middleton, "Plug-and-play network reconfiguration algorithms to maintain regularity and low network reconfiguration needs," *IEEE Control Systems Letters*, vol. 6, pp. 3451–3456, 2022.

[12] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[13] Y.-Q. Xia, Y.-L. Gao, L.-P. Yan, and M.-Y. Fu, "Recent progress in networked control systems — a survey," *International Journal of Automation and Computing*, vol. 12, no. 4, pp. 343–367, 2015.

[14] K. Fan, "On a theorem of weyl concerning eigenvalues of linear transformations i," *Proceedings of the National Academy of Sciences*, vol. 35, no. 11, pp. 652–655, 1949.

[15] F. Harary, *Graph theory*. Addison-Wesley series in mathematics, Reading, Mass.: Addison-Wesley, 4. print ed., 1995.

[16] H. P. Williams, *Model building in mathematical programming*. John Wiley & Sons, 2013.

[17] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *In Proceedings of the CACSD Conference*, 2004.