A Networked Transferable Belief Model approach for Distributed Data Aggregation

Andrea Gasparri, Flavio Fiorini, Maurizio Di Rocco and Stefano Panzieri.

***Abstract*—This works focus on the extension of the Transferable Belief Model (TBM) to a multi-agent distributed context where no central aggregation unit is available and the information can be exchanged only locally among agents. In this framework, agents are assumed to be independent reliable sources which collect data and collaborate to reach a common knowledge about an event of interest. Two different scenarios are considered: in the first one agents are supposed to provide observations which do not change over time (static scenario), while in the second one agents are assumed to dynamically gather data over time (dynamic scenario). A protocol for distributed data aggregation, which is proved to converge to the basic belief assignment (BBA) given by an equivalent centralized aggregation schema based on the TBM, is provided. Since multi-agent systems represent an ideal abstraction of actual networks of mobile robots or sensor nodes, that are envisioned to perform the most various kind of tasks, we believe the proposed protocol paves the way to the application of the TBM in important engineering fields such as multi-robot systems or sensor networks, where the distributed collaboration among players is a critical and yet crucial aspect.**

***Index Terms*—Transferable Belief Model, Distributed Algorithms, Multi-Agent Systems**

I. Introduction

Data fusion is a research area that is growing rapidly due to the fact that it provides means for combining pieces of information coming from different sources/sensors. As a result, an enhanced overall system performance, i.e., improved decision making, increased detection capabilities, diminished number of false alarms, improved reliability, with respect to separate sensors/sources can be achieved [1].

Indeed, data fusion techniques play an important role in the context of multi-agent systems where information coming from different sources must be aggregated in order to provide a meaningful description of the surrounding environment. The majority of works available in the literature are based on the Bayesian framework, where the aggregation is achieved by applying the Bayes rule. The most representative example is the Kalman Filter, where noisy data is assumed to be described by means of a Gaussian probability distribution [2]. Nevertheless, several works have been proposed to deal with the multi-agent data fusion in a Bayesian framework [3], [4], [5], [6]. Indeed, as suggested in [3], the single-agent paradigm might be inadequate when uncertain reasoning is performed by entities of a system between which there is some distance, either spatial, temporal or semantic. For these systems, a multi-agent view, where each agent is an autonomous intelligent subsystem, is thus more suitable. Each agent holds its own partial domain knowledge, accesses some external information source and consumes some computational resource. Bayesian Networks represent a possible way of implementing this view within the Bayesian Framework. A Bayesian Networks is a directed acyclic graph where nodes represent variables and the graph represents conditional independence relations among the variables. The reader is referred to [7] for further details.

The Theory of Evidence introduced by Arthur P. Dempster and Glenn Shafer (DS) represents a valid alternative to the Bayesian framework [8]. The main difference concerns the way in which the ignorance is handled: in the probabilistic framework the uncertainty is treated by splitting the amount of credibility among plausible events, in the DS framework a belief is assigned to the set describing all the plausible hypotheses without supporting any in particular. Several works can be found in the literature providing a comparison between these two frameworks [9], [10], [11], [12]. Depending on the specific application, one framework can be more adequate than the other [13].

The DS framework was further extended by the Transferable Belief Model (TBM) introduced by Philippe Smets [14]. In particular, TBM introduces the idea of open word assumption in the DS framework. This implies the set of hypotheses not to be exhaustive, therefore information can be contradictory. Indeed, the concept of contradiction is a powerful tool to detect cases where information fusion has to be considered unreliable, case that is not considered in the Bayesian technique. The main limitation of this framework is the computational complexity, which grows exponentially with respect to the number of elements. To overcome this drawback, several approximation techniques have been proposed [15], [16]. However, in case a minimal number of events is enough to model the problem, the TBM approach has been effectively used, e.g, in diagnostic applications [17] and target identification [18].

In this paper the data aggregation problem for a multi-agent system is investigated. A multi-agent system represents an ideal abstraction of actual networks of mobile robots or sensor nodes that are envisioned to perform the most various kind of tasks. In the last decade networked multi-agent systems have drawn the attention of a large part of the research community [19], [20], [21], [22]. The motivation behind the interest on multi-agent systems is that a multi-agent approach offers several advantages such as a larger range of task domains or a higher robustness and flexibility [23]. On the other hand, the inherently distributed nature of these systems makes the design of effective algorithms very challenging as the overall

All the authors are with the Department of Computer Science and Automation, University of “Roma Tre”, Roma, 00146 Italy Corresponding Author: *Andrea Gasparri* Email: *gasparri@dia.uniroma3.it*

performance depends significantly on issues arising from the complex interactions among the agents [24]. In this work, we attempt to answer to the following question: Is there a way to extend the TBM framework to a distributed context where no central aggregation unit is available and the information can be exchanged only locally? Indeed, being able to answer such a question would make it possible to effectively apply the TBM in important engineering fields such as multi-robot systems or sensor networks, where the distributed collaborations among players is a critical and yet crucial aspect. In this framework, agents are assumed to be independent reliable sources which collect data and collaborate to reach a common knowledge. Two different scenarios are considered, in the first one agents are supposed to provide observations which do not change over time (static scenario), while in the second one agents are assumed to dynamically gather data over time (dynamic scenario). A protocol for distributed data aggregation which is proved to converge to the basic belief assignment (BBA) given by a centralized aggregation based on the Transferable Belief Model (TBM) conjunctive rule is provided.

Some works can be found in the literature discussing either static and dynamic aggregation, [25], [26], [27], [28]. In [25], the problem of sensors reliability estimation is addressed. The idea is to introduce a metric distance between the sensor readings and the reality. This allows to tune the discounting factor belonging to each sensor in order to minimize the distance itself. Both cases where the training data set is deterministic or prone to uncertainties are considered. Regarding the uncertainties scenario, an elegant comparison between standard (Bayesian) classifier and the proposed approach is shown. Furthermore, an extension to data-fusion is provided: in this case, an optimization process, to find out which is the optimal discounting distribution among the sources, is adopted by considering the pignistic probability achieved after applying the conjunctive rule on the observed belief assignments. In [26], the authors propose a novel framework for evaluating sensor reliability by integrating prior and contextual information. Differently from the previous case, where the knowledge about the training data set was expressed by means of pignistic probabilities, in this work this knowledge is represented by means of belief functions. Consequently, authors use a dissimilarity metric, namely the evidence distance, which guarantees the respect of the mathematical properties holding in the TBM framework. Relying on these assumptions, analytical procedures allow to achieve discounting factors. The proposed approach aims at finding an estimate of both data-fusion quality and sensor reliability. In the latter case, the discounting factor is gathered using the Shapley entropy about belief functions as an indicator. In [27], the authors consider uncertain data whose uncertainty is represented by belief functions whose combination can be partially conflictual. In particular, the authors discuss the nature of the combinations (conjunctive versus disjunctive, revision versus updating, static versus dynamic data fusion), they argue about the need for a normalization, examine the possible origins of the conflicts, determine if a combination is justified and analyze many of the proposed solutions. Furthermore, the authors discuss the *Markovian decaying assumption*, a representation that allows to model the decaying of the source reliability over the time: this model is a useful representation when dynamic systems are taken into account. In [28], a framework for fault diagnosis is described. In particular the Dynamic Evidential Reasoning (DER) is presented: it allows to model a system by a set of attributes. These attributes, whose origin can be heterogeneous, are weighted by discounting factors and merged in order to provide a one-step forward estimate about the fault-state of the system. The evolution of the system relies on the Markovian decay assumption which allows to cope with dynamical system faults. In order to properly set the initial attributes and the related importance into the model, an optimization process over a training data-set is exploited by dedicated optimization tools. The framework fits well in time-varying examples like two-tanks system and gyroscope reliability.

Compared to these contributions, our work is mainly focused on the networking aspects concerning the data aggregation rather than the interpretation of the data itself. More specifically, our main contribution is the development and the theoretical characterization of a distributed protocol for data aggregation within the TBM framework in a multi-agent context.

II. Theory of Evidence

The Theory of Evidence is a formalism which can be used for modeling uncertainty instead of classical probability. Theory of Evidence embraces the familiar idea of using a number between zero and one to indicate the degree of confidence for a proposition on the basis of the available evidence.

Let $\Omega = \{\omega_1, \ldots, \omega_n\}$ be a finite set of possible values of a variable ω, where the elements ω_i are assumed to be mutually exclusive. Let $\Gamma(\Omega) \triangleq 2^\Omega = \{\gamma_1, \ldots, \gamma_{|\Gamma|}\}$ be the power set associated to it. In this framework, the interest is focused in quantifying the confidence of propositions of the form: "the true value of ω is in γ", with $\gamma \in \Gamma$. The propositions of interest are therefore in one-to-one correspondence with the subset Ω, and the set of all propositions of interest corresponds to the elements of Γ. The set Ω so defined, is referred to as *frame of discernments*.

***Definition** 1 (BBA):* A function $m : 2^\Omega \to [0, 1]$ is called a basic belief assignment if

$$\sum_{\gamma_a \in \Gamma} m(\gamma_a) = 1 \quad \text{with} \quad m(\emptyset) = 0. \tag{1}$$

Thus for $\gamma_a \in \Gamma$, $m(\gamma_a)$ is the part of confidence that supports exactly γ_a, i.e. the fact that the true value of ω is in γ_a, but due to the lack of further information, does not support any strict subset of γ_a. The first condition reflects that the total confidence has measure one and the second condition reflects that the total confidence has measure one. Note that $m(\gamma_a)$ and $m(\gamma_b)$ can be both equal to zero even if $m(\gamma_a \cup \gamma_b) \neq 0$. Further, $m(\cdot)$ is not monotone under inclusion, i.e. $\gamma_a \subset \gamma_b$ does not imply $m(\gamma_a) < m(\gamma_b)$.

Notice that the BBA represents the atomic information in the theory of evidence.

The main criticism to Shafer formulation concerns the application of the Dempster-Shafer (DS) combination rule. In fact, whenever there is a strong conflict between sources to be combined, the straightforward application of DS combination rule can produce a result in which certainty is assigned to the minority opinion [29]. A more refined approach is based on the Transferable Belief Model (TBM) proposed by Philips Smets in [30]. The TBM theory, like the Shafer formulation, relies on the concept of basic belief assignment function, but removes the assumption of $m(\emptyset) = 0$. This allows to omit the normalization constant in the Dempster's rule of combination and conditioning.

***Definition** 2 (Smets - operator ⊗):*
In the TBM, the combination rule is, therefore, defined in this way:

$$s_{ij} \triangleq s_i \otimes s_j = \Big\{ (m_i \otimes m_j)(\gamma_a);\ \gamma_a \in\ \Gamma \Big\} \tag{2}$$

where:

$$m_{ij}(\gamma_a) \triangleq (m_i \otimes m_j)(\gamma_a) = \sum_{\substack{\gamma_b,\, \gamma_c \\ \gamma_b \cap \gamma_c = \gamma_a}} m_i(\gamma_b) m_j(\gamma_c). \tag{3}$$

Note that, for the sake of readability the notation $m_i(\gamma_a) \otimes m_j(\gamma_a) \triangleq (m_i \otimes m_j)(\gamma_a)$ will be used indiscriminately in the rest of the paper. In the same way, the notation $s_{ijk} = s_i \otimes s_j \otimes s_k$ will be used to compactly denote several Smets aggregations.

The fact that $m(\emptyset) > 0$ can be explained in two ways: the open world assumption and the quantified conflict. The open world assumption reflects the idea that Ω might not be exhaustive, i.e. it might not contain all the possibilities. Under this interpretation, being $\emptyset$ the complement of Ω, the mass $m(\emptyset) > 0$ represent the modeling errors, that is the fact that the truth might not be contained in Ω. The second interpretation of $m(\emptyset) > 0$ is that there is some underlying conflict between the sources that are combined in order to produce the BBA m. Hence, the mass assigned to $m(\emptyset)$ represents the degree of conflict. In particular, it can be computed as follows:

$$m_{ij}(\emptyset) = 1 - \sum_{\substack{\gamma_a \neq \emptyset \\ \gamma_a \in \Gamma}} m_{ij}(\gamma_a) \tag{4}$$

III. Problem Setting

Let us consider a network of agents described by an undirected graph $\mathcal{G} = \{V, E\}$, where $V = \{v_i : i = 1, \ldots, n\}$ is the set of nodes (agents) and $E = \{e_{ij} = (v_i, v_j)\}$ is the set of edges (connectivity) representing the point-to-point communication channel availability. Note that, since the graph is undirected, the existence of the edge e_{ij} (from node i to node j) implies the existence of the edge e_{ji} (from node j to node i). Thus, no distinction will be made in the following between e_{ij} and e_{ji}. Moreover, we will refer to $\mathcal{N}(i)$ as the neighborhood of agent i, namely the set of indices of the agents directly connected through an edge with agent i. In this framework an agent can be of any kind, for instance a mobile robot, a sensor node, a software agent or else.

Let us now introduce a simple classification problem to explain the issues which arise from the straightforward application of the TBM within a distributed context. In particular, the following assumptions are made for this scenario: 1) no central unit is available for data aggregation, 2) communication among sensor nodes is limited to one-hop neighbors. Indeed, these assumptions reflect the operational conditions for the majority of multi-robot and sensor network applications [31], [32], [33]. Furthermore, it should be noticed that we do not focus on problems related to the data interpretation such as the reliability problem addressed in [25]. Differently, our attention is towards the issues which arise from the extension of the TBM within a distributed context where only local communication is permitted and no central aggregation unit is available.

Set #	Node 1	Node 2	Node 3	$s_1 \otimes s_2 \otimes s_3$
$\emptyset$	0	0	0	0.79
$\{a\}$	0.2	0.8	0.1	0.066
$\{b\}$	0.5	0.1	0.85	0.1425
$\{a, b\}$	0.3	0.1	0.05	0.0015

TABLE I
AN EXAMPLE OF CLASSIFICATION PROBLEM. SENSOR NODES BBA.

As an example, similarly to the classification problem proposed in [25], let us consider two different possible classes of targets, e.g., cars or trucks, that is $\Omega = \{a, b\}$. Let us supposed that the aggregated sensor readings provided by the nodes on the classes are expressed by the BBAs $s_i = \Big\{m_i(\emptyset), m_i(a), m_i(b), m_i(a, b)\Big\}$ given in Table I. In particular, let us assume each node to be equipped with some sensors such as a speed sensor, a volume sensor and so on.

The objective of the sensor network is to perform this classification task by means of a distributed data fusion based on point-to-point communication over the network topology depicted in Figure 1. Let us now investigate the problems arising from the application of the Smets combination rule. In particular, let us suppose that node 1 first collaborates with node 2 and successively sets up a collaboration with node 3. At this point, a question arises: what happens if node 1 collaborates again with node 2? Let us further investigate this situation.

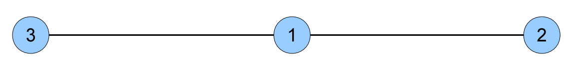


Fig. 1. Network topology.

In detail, when node 1 collaborates with node 2, they perform a data fusion over their current BBAs, that is $s_{12} = s_1 \otimes s_2 = \{0.42,\, 0.42,\, 0.13,\, 0.03\}$. As a result, both nodes reach a new common knowledge about the class of the target. In the same way, when node 1 collaborates with node 3, they perform a data fusion over their current BBAs and reach a new (common) knowledge about the class of the target which classifies the target as a truck, that is $s_{123} = s_{12} \otimes s_3 = \{0.79,\, 0.066,\, 0.1425,\, 0.0015\}$. Indeed, this is the same knowledge ($s_{123} = s_1 \otimes s_2 \otimes s_3$) that would be reached in a centralized scenario (Table I). At this point, if node 1 and node 2 collaborate again, by using the Smets combination rule, they would reach a new wrong knowledge which classifies the target as a car,namely $s_{12312} = s_{123} \otimes s_{12} = \{0.91, 0.0606, 0.02865, 0.00015\}$), where the information due to the first communication among the two nodes would be considered twice. Therefore, the Smets operator cannot be used in a distributed context as the result of the target classification depends upon the particular sequence of selected edges. Indeed, a different combination strategy must be designed in order to overcome this limitation.

The key idea of this work is that the current knowledge of a node can be split with respect to any of its neighbors in two parts. The first represents the portion of information shared by the current BBAs of the two nodes, while the second is the novel portion of information brought by the BBA of each node w.r.t. another one. As a result, this issue can be simply overcome by restricting the aggregation among nodes to the novel portion of information, and then combining the obtained result with the shared knowledge. In this way, the result of their previous aggregation is not considered twice by node 1 and 2 in the next collaboration. A formalization of this idea is provided in the first part of the paper.

Let us now further suppose that an agent can dynamically collect new observations over time. As a consequence, the following question arises: what happens if some agents update their observations while performing the data aggregation? Indeed, the protocol must take into account the fact that a node might update its observation. To this end, a proper extension of the proposed distributed data aggregation protocol to overcome this limitation is discussed in the second part of the paper.

IV. Distributed Data Aggregation Via Networked TBM - static scenario

In this section, a local interaction rule $\mathcal{R}$ to perform the distributed TBM data aggregation within a multi-agent system is described. In particular, each agent is supposed to provide the same observation over time. Furthermore, the following assumptions on the network of agents are made:

***Assumptions** 1:*

- The network can be described by a connected undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$.
- The communication range is limited by a maximum communication radius r.
- The communication among agents is asynchronous, *gossip* like [34].
- A distributed algorithm to build a spanning tree $\mathcal{T} = \{V, \hat{E}\}$ with $\hat{E} \subseteq E$ is available to the agents, for example by using [35], [36].
- Agents are capable to handle the storage of a proper set of data.

In the proposed framework the interaction among agents can be formally modeled by means of a gossip algorithm [34] is defined as a triplet $\{\mathcal{S}, \mathcal{R}, \mathfrak{e}\}$ where:

- $\mathcal{S} = \{s_1, \ldots, s_n\}$ is the set containing the local state $s_i \in \mathbb{R}^q$ of each agent i in the network.
- $\mathcal{R}$ is the local interaction rule ($\oplus$ binary operator) that, for any couple of agents (i, j) with $e_{ij} \in E$, gives:

$$\mathcal{R}: \quad \mathbb{R}^q \times \mathbb{R}^q \longrightarrow \mathbb{R}^q,$$

  with q the number of focal elements and $\mathcal{R}^q$ the set of all possible mass assignment vectors.
- $\mathfrak{e}$ is the edge selection process that specifies which edge $e_{ij} \in E(t)$ is selected at time t.

From an algorithmic point of view, a possible implementation of the gossip algorithm is given in Algorithm 1.

Algorithm 1: Gossip Algorithm

Data: $t = 0,\ s_i(0) \quad \forall\, i = 1, \ldots, n$.
Result: $s_i(t_{stop}) \quad \forall\, i = 1, \ldots, n$.

while *stop_condition* **do**
- Select an edge $e_{ij} \in E(t)$ according to $\mathfrak{e}$.
- Update the states of the selected agents applying $\mathcal{R}$:

$$\begin{array}{rcl} s_i(t+1) & = & s_i(t) \oplus s_j(t) \\ s_j(t+1) & = & s_j(t) \oplus s_i(t) \end{array}$$

- Let $t = t + 1$.

end

Now follows the definition of $\mathcal{S}$ and $\mathcal{R}$:

***Definition** 3 ($\mathcal{S}$):* Let $\mathcal{S}(t) = \{s_1(t), \ldots, s_n(t)\}$ be the set of the agents states defined with respect to a finite frame of discernment $\Omega = \{\omega_1, \ldots, \omega_m\}$, where $s_i(t) = \{m_i(t, \gamma_a),\ \gamma_a \in \Gamma\},\ s_i(t) \in \mathbb{R}^{|\Gamma|}$ is the set of basic belief assignment (BBA) of agent i over the power set $\Gamma(\Omega)$ at a given time $t \in \mathbb{N}$. Note that, in the following the time dependence will be omitted for sake of clarity if not strictly required.

Let us now introduce the binary operator $\odot$, which is useful to break up any set of basic belief assignment with respect to any other one.

***Lemma** 1:* Let us consider two sets of basic belief assignments (BBA) $s_k = \big\{m_k(\gamma_a);\quad \forall \gamma_a \in \Gamma\big\}$, and $s_i = \big\{m_i(\gamma_a);\quad \forall \gamma_a \in \Gamma\big\}$ for which the following holds: $s_k = s_i \otimes s_j$. It can be defined an operator $\odot$:

$$\tilde{s}_k^i \triangleq s_k \odot s_i = s_j. \tag{5}$$

In particular, each element of $s_j = \big\{m_j(\gamma_a);\quad \forall \gamma_a \in \Gamma\big\}$, can be computed recursively as follows:

$$m_j(\gamma_a) = \frac{m_k(\gamma_a) - \sum_{\substack{\gamma_b \cap \gamma_c = \gamma_a \\ \gamma_b \supset \gamma_a}} m_j(\gamma_b) m_i(\gamma_c)}{\sum_{\gamma_a \subseteq \gamma_b} m_i(\gamma_b)}, \tag{6}$$

by starting from the element of the power-set with highest cardinality, $\gamma_{|\Gamma|} = \{\Omega\}$, and moving down to the elements with cardinality equal to one, i.e., $\{\gamma_{i+1} = \{\omega_i\},\ i = 1, \ldots, n\}$, with $\gamma_1 = \emptyset$.

Proof: The proof is a simple consequence of the application of Smets operator $\otimes$. Let us assume s_k can be written as the Smets aggregation of s_i and s_j:

$$s_k \triangleq s_i \otimes s_j = \Big\{(m_i \otimes m_j)(\gamma_i);\ \gamma_i \in\ \Gamma\Big\}$$

where:

$$m_k(\gamma_a) \triangleq (m_i \otimes m_j)(\gamma_a) = \sum_{\gamma_b \cap \gamma_c = \gamma_a} m_i(\gamma_b) \cdot m_j(\gamma_c) = m_j(\gamma_a) \sum_{\gamma_a \subseteq \gamma_b} m_i(\gamma_b) + \sum_{\substack{\gamma_b \cap \gamma_c = \gamma_a \\ \gamma_b \supset \gamma_a}} m_j(\gamma_b) m_i(\gamma_c).$$

At this point, by collecting with respect to $m_j(\gamma_a)$ the following expression is obtained:

$$m_j(\gamma_a) = \frac{m_k(\gamma_a) - \sum_{\substack{\gamma_b \cap \gamma_c = \gamma_a \\ \gamma_b \supset \gamma_a}} m_j(\gamma_b) m_i(\gamma_c)}{\sum_{\gamma_a \subseteq \gamma_b} m_i(\gamma_b)}.$$

Therefore, $s_j = \{m_j(\gamma_a); \quad \forall \gamma_a \in \Gamma\}$ is obtained. ■

Let us now introduce the operator $\mathcal{R}$ which is used in the static scenario to perform the local aggregation among the agents.

***Definition** 4 ($\mathcal{R}$ - operator $\oplus$):*
Let $\mathcal{R}$ be a rule to combine the basic belief assignments for two agents (i, j) such that $e_{ij} \in \hat{E}$. This rule can be defined from the agent i standpoint (specularly for agent j) as follows:

$$\begin{aligned} s_i(t+1) &= s_i(t) \oplus s_j(t) \\ &= \Big\{ \big(\tilde{m}_i^j(t,\gamma_a) \otimes \tilde{m}_j^i(t,\gamma_a)\big) \otimes \bar{m}_{i,j}(t,\gamma_a); \\ &\quad \forall \, \gamma_a \in \Gamma \Big\}, \end{aligned} \tag{7}$$

with $\otimes$ the Smets operator. Let us denote with $\tilde{s}_i^j(t) = \Big\{\tilde{m}_i^j(t,\gamma_a); \quad \forall \gamma_a \in \Gamma\Big\}$ the novelty of the agent i with respect to the agent j, which can be computed recursively as follows:

$$\tilde{m}_i^j(t,\gamma_a) = \frac{m_i(t,\gamma_a) - \sum_{\substack{\gamma_b \cap \gamma_c = \gamma_a \\ \gamma_b \supset \gamma_a}} \tilde{m}_i^j(t,\gamma_b) \bar{m}_{i,j}(t,\gamma_c)}{\sum_{\gamma_a \subseteq \gamma_b} \bar{m}_{i,j}(t,\gamma_b)} \tag{8}$$

and $\bar{s}_{i,j}(t) = \{\bar{m}_{i,j}(t,\gamma_a); \quad \forall \gamma_a \in \Gamma\}$ (or equivalently $\bar{s}_{j,i}(t)$) is the common knowledge, i.e., the knowledge stored by both agents after their last aggregation, set to the neutral element $\mathfrak{n} = \{0, 0 \, \dots, 0, 1\}$ of the TBM conjunctive rule before their first aggregation.
Note that, as a consequence of Lemma 1, for a given agent i the following relation holds between the novelty and the common knowledge with any other agent j:

$$\begin{aligned} s_i(t) &= \Big\{ m_i(t,\gamma_a); \quad \forall \gamma_a \in \Gamma \Big\} = \tilde{s}_i^j(t) \otimes \bar{s}_{ij}(t) \\ &= \Big\{ \tilde{m}_i^j(t,\gamma_a) \otimes \bar{m}_{i,j}(t,\gamma_a); \quad \forall \gamma_a \in \Gamma \Big\} \end{aligned} \tag{9}$$

Furthermore, for any couple of agents (i, j), the related states s_i and s_j are equal if and only if they are completely described by their common knowledge, i.e. $s_i = s_j = \bar{s}_{i,j}$.

***Remark** 1:* A few important remarks are now in order:
1) In order to apply the local interaction rule $\mathcal{R}$, an agent must have stored all the most recent collaborations with its neighbors, that is $\Big\{s_i \oplus s_j; \quad j \in \mathcal{N}(i)\Big\}$.
2) As only information concerning collaborations among (1-hop) neighbors are required, the algorithm is fully distributed and scalable in terms of memory requirement with respect to the size of the network.

Regarding the knowledge decomposition (operator $\odot$) and local aggregation rule (operator $\oplus$) proposed in this work, a similarity can be noticed with the concept of *Dempsterian Specialization Matrix* described in [37] and its application to the combination of distinct pieces of evidence. In this work, Smets presents a tentative definition of the concept of distinctness for two pieces of evidence. To this end, by assuming two BBAs s_A and s_B to be two updating of a third BBA s_0, Smets provides a technique, based on the concept of specialization matrix, to decompose the knowledge s_{AB} resulting from the combination of these two BBAs (s_A and s_B). In particular, by noticing that such a decomposition takes into account the BBA s_0, Smets argues that the two BBAs s_A and s_B can be considered distinct only if the BBA s_0 is vacuous, i.e., $m_0(\Omega) = 1$. Although the focus of the work proposed by Smets is completely different, from a conceptual perspective, there is a similarity between the notion of novelty and common knowledge adopted in our work, and the key elements involved in the knowledge decomposition proposed by Smets. The reader is referred to [37] for further details.

At this point, in order to prove the convergence of the proposed algorithm, some properties concerning the local interaction rule $\mathcal{R}$ must be introduced.

***Lemma** 2 ($\mathcal{R}$ properties):* The local interaction rule $\mathcal{R}$ defined according to eq. (7) has the following properties:

$$\begin{aligned} s_i \oplus s_j &= s_j \oplus s_i && \text{(commutativity)} \\ s_i \oplus s_j &= s_i \quad \text{if } \bar{s}_{i,j} = s_i && \text{(idempotence)} \\ (s_i \oplus s_j) \oplus s_k &= s_i \oplus (s_j \oplus s_k) && \text{(associativity)} \end{aligned} \tag{10}$$

for each triple $(i,\, j,\, k) : e_{ij}, e_{jk} \in \hat{E}$.

Proof: The properties can be proven by applying the definition given in eq. (7).

- Commutativity:
  Let us consider two agents (i, j), then from Definition 4 we have:
  $$s_i \oplus s_j = \Big(\tilde{s}_i^j \otimes \tilde{s}_j^i\Big) \otimes \bar{s}_{i,j} = \Big(\tilde{s}_j^i \otimes \tilde{s}_i^j\Big) \otimes \bar{s}_{j,i} = s_j \oplus s_i$$
  where $\Big(\tilde{s}_i \otimes \tilde{s}_j\Big) = \Big(\tilde{s}_j \otimes \tilde{s}_i\Big)$ comes from the commutativity property of the Smets operator $\otimes$ and $\bar{s}_{i,j} = \bar{s}_{j,i}$ by definition.
- Idempotence:
  Let us consider two agents (i, j) for which $s_i = s_j = \bar{s}_{i,j}$ at a given time t, then we have:
  $$s_i \oplus s_j = \Big(\tilde{s}_i^j \otimes \tilde{s}_j^i\Big) \otimes \bar{s}_{i,j} = \Big(\mathfrak{n} \otimes \mathfrak{n}\Big) \otimes \bar{s}_{i,j} = \bar{s}_{i,j} = s_i$$
- Associativity:
  Let us consider a triplet of agents $(i,\, j,\, k)$ such that $e_{ij}, e_{jk} \in \hat{E}$, then we have:
  $$\begin{aligned} \Big(s_i \oplus s_j\Big) \oplus s_k &= s_{ij} \oplus s_k &&= \tilde{s}_{ij}^k \otimes \tilde{s}_k^{ij} \otimes \bar{s}_{ij,k} \\ &= s_{ij} \otimes \tilde{s}_k^{ij} &&= \tilde{s}_i^j \otimes s_j \otimes \tilde{s}_k^{ij} \\ &= \tilde{s}_i^j \otimes s_j \otimes \tilde{s}_k^j &&= \tilde{s}_i^j \otimes s_{jk} \\ &= \tilde{s}_i^{jk} \otimes s_{jk} &&= \tilde{s}_i^{jk} \otimes \tilde{s}_{jk}^i \otimes \bar{s}_{ij,k} \\ &= s_i \oplus s_{jk} &&= s_i \oplus \Big(s_j \oplus s_k\Big) \end{aligned}$$

where the equivalent relations $\tilde{s}_k^{ij} = \tilde{s}_k^{j}$ and $\tilde{s}_i^{j} = \tilde{s}_i^{jk}$ come from the independence of knowledge between nodes i and k with respect to j, due to the properties of the topology structure of the communication graph, i.e., a spanning-tree $\mathcal{T}$. ■

So far, we have introduced a local interaction rule $\mathcal{R}$ and we have described its properties. In the following, it will be shown that if the agents apply the gossip algorithm described in Sec. III with such a local interaction rule $\mathcal{R}$ over the spanning-tree $\mathcal{T}$, they converge toward a common BBA. In particular, it will be shown that such a BBA is the same as in the case of a centralized aggregation based on TBM conjunctive rule [30].

***Definition** 5 (Centralized TBM):* Let us consider a system of n agents (sources) where each agent i provides an independent set of observations at time $t = 0$ described by the basic belief assignment $s_i(0) = \{m_i(0, \gamma_a);\ \gamma_a \in \Gamma\}$. A centralized aggregation schema would provide the following aggregated BBA:

$$s_{1\,2\,\dots\,n} = s_1 \otimes s_2 \otimes \dots \otimes s_n \tag{11}$$

Let us now introduce the concept of a time-dependent forest $\mathcal{F}(t, t+\Delta)$ with respect to a given tree $\mathcal{T}$ over time as follows:

***Definition** 6:* Let us define $\mathcal{F}(t, t+\Delta t) = \{V, \hat{E}(t, t+\Delta t)\}$, with $\hat{E}(t, t+\Delta t) = \bigcup_{z=t}^{t+\Delta t} \mathfrak{e}(z)$ and $\mathfrak{e}(z) \in \hat{E}$, as the forest resulting from the union of all the edges given by the edge selection process over the set $\hat{E}$ from time t to time $t + \Delta t$. Obviously, if the edge process $\mathfrak{e}$ is such that in the time interval $(t, t+\Delta t)$ the forest $\mathcal{F}(t, t+\Delta t)$ is connected, then the spanning tree $\mathcal{T}$ is obtained.

In order to prove the main result of the paper, a useful relationship between the Smets operator $\otimes$ and the proposed interaction rule $\oplus$ is now introduced.

***Lemma** 3:* Let us consider three agents (i, j, k) such that $e_{ij}, e_{jk} \in \hat{E}$ with their observations $s_i(0), s_j(0), s_k(0)$ at time $t = 0$. The following holds:

$$s_i(0) \otimes s_j(0) \otimes s_k(0) = s_i(0) \oplus s_j(0) \oplus s_k(0) \tag{12}$$

Proof: The lemma can be proven by applying the definition given in eq. (7) along with the properties given in eq. (9):

$$\begin{aligned} s_i(0) \oplus s_j(0) \oplus s_k(0) &= \Big(s_i(0) \oplus s_j(0)\Big) \oplus s_k(0) \\ &= \Big(\tilde{s}_i^{j}(0) \otimes \tilde{s}_j^{i}(0) \otimes \bar{s}_{i,j}(0)\Big) \oplus s_k(0) \\ &= \Big(s_i(0) \otimes s_j(0) \otimes \mathfrak{n}\Big) \oplus s_k(0) \end{aligned}$$

where $\tilde{s}_i^{j}(0) \otimes \tilde{s}_j^{i}(0) \otimes \bar{s}_{i,j}(0) = s_i(0) \otimes s_j(0) \otimes \mathfrak{n}$ is due to the independence of the agents observation. Now, by defining $s_z(0) = s_i(0) \otimes s_j(0)$, it follows that:

$$\begin{aligned} s_z(0) \oplus s_k(0) &= \tilde{s}_z^{k}(0) \otimes \tilde{s}_k^{z}(0) \otimes \bar{s}_{z,k}(0) \\ &= s_z(0) \otimes s_k(0) \otimes \mathfrak{n} \\ &= s_i(0) \otimes s_j(0) \otimes s_k(0) \end{aligned}$$

■

Let us now introduce the main result of the paper, that is the convergence of the proposed gossip algorithm towards the basic belief assignment (BBA) as in the centralized aggregation schema given in Definition 5.

***Theorem** 1 (Distributed TBM):* Let us consider a gossip algorithm $\{\mathcal{S}, \mathcal{R}, \mathfrak{e}\}$ over a spanning-tree $\mathcal{T} = \{V, \hat{E}\}$ with $\mathcal{S}$ and $\mathcal{R}$ defined respectively as in Definition 3 and Definition 4. Let us assume each agent i at time t= 0 provides an independent set of observations described by the basic belief assignment $s_i(0) = \{m_i(0, \gamma_a);\ \gamma_a \in \Gamma\}$. If $\mathfrak{e}$ is such that $\forall\, t \quad \exists\, \Delta t \in \mathbb{N}$ so that the time-variant forest $\mathcal{F}(t, t+\Delta t)$ is connected, then there will exist a time $t = \bar{t}$ so that:

$$s_i(t') = s_1(0) \otimes s_2(0) \otimes \dots \otimes s_n(0) \quad \forall\, t' > \bar{t}, \tag{13}$$

that is, each agent i converges toward the same BBA as in the centralized aggregation schema given in Definition 5.

Proof: The proof of the theorem consists of three steps. First, it will be proven that a steady-state exists for the proposed gossip algorithm. Successively, it will be proven that such an algorithm always converges toward a steady-state. Finally, it will be proven that the steady-state is unique and it is the same as the result of the centralized aggregation schema given in eq. (11).

1) Steady-State Existence: In order to prove the existence of a steady-state for the proposed gossip algorithm, it will be shown that a sufficient and necessary condition is that all the agents share the same state $\bar{s}$. In fact, if all the agents have the same state $\bar{s}$, according to the interaction rule given in eq. (7), they will always send the neutral element $\mathfrak{n}$ for any further aggregation. Therefore, the state $\bar{s}$ is itself a steady state for the multi-agent system. Furthermore, let us prove by contradiction this condition to be necessary as well. To this end, let us consider a spanning-tree $\mathcal{T}$ computed by the agents in a distributed fashion. Now, let us suppose two agents i and j have reached two different steady states over the network, that is $s_i(t) = s'$ and $s_j(t) = s''$. Therefore, according to the definition of a spanning-tree, there will always exist a (unique) path connecting the two nodes i and j. Let us now consider for such a spanning-tree $\mathcal{T}$ the path $p_{ij} = \{v_i,\ v_{k \in \mathcal{N}_i},\ \dots,\ v_{h \in \mathcal{N}_j},\ v_j\}$ connecting these two agents i and j. In particular, as agent i has reached the state s', its neighbor k will always send to it the neutral element $\mathfrak{n}$ as novelty for any further aggregation. This implies that, the agent k must have reached itself the same steady state s' and be receiving the neutral element $\mathfrak{n}$ by its neighbors. The same argument can be applied to the agent j and its neighbor h with respect to the steady state s''. Now, by iterating this

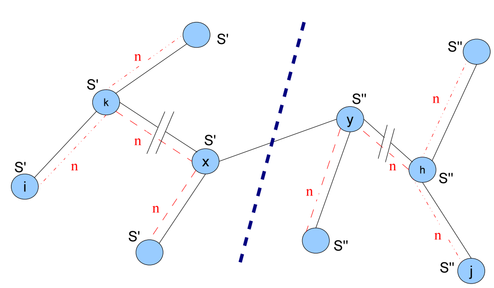


Fig. 3. Novelty contraction over the spanning tree $\mathcal{T}$ at different time-interval.

reasoning from both ends of the path there will be a cut where all the nodes on a side will have reached the same steady-state s' as agent i, while on the other side all the agents will

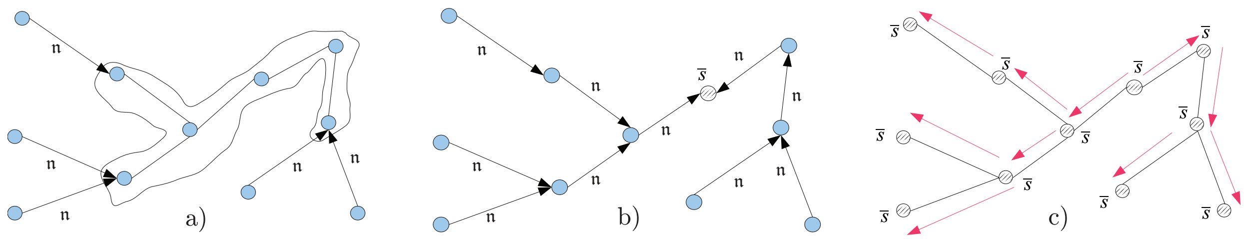


Fig. 2. Steady-state convergence over the spanning tree $\mathcal{T}$ at different time-interval: a) after the first interval of time during which the tree $\mathcal{T}$ is obtained, the related leaves will send only the neutral element b) after a certain amount of time the root of $\mathcal{T}$ achieve the steady state $\bar{s}$ c) Finally the steady state is spread over the whole net

have reached the same steady-state s'' as agent j, as shown in Figure 3. Let us call x and y the two agents on the boundaries of the cut. Since, x and y have both reached a steady state, s' and s'' respectively, they must be sending the neutral element $\mathfrak{n}$ as novelty to each other. However, the two steady states s' and s'' have been supposed to be different, therefore the two agents x and y *cannot* be sending the neutral element $\mathfrak{n}$ to each other. Indeed, this would be possible only if the two steady states s' and s'' were the same steady state $\bar{s}$, which gives the absurd. Therefore a steady state holds if and only if $s_i = s_j \quad \forall i, j \in N$.

2) Steady-State Convergence: In order to prove the convergence of the proposed algorithm towards a steady-state, let us consider a spanning tree $\mathcal{T}$ computed by the agents in a distributed fashion over the network topology $\mathcal{G}$, as shown in Figure 2. Now, let us consider an interval of time $[t_0, t_0+\Delta t_0]$ for which the forest $\mathcal{F}(t_0, t_0+\Delta t_0)$ is connected. This implies that some agents play the role of leaves for the resulting spanning-tree $\mathcal{T}$. According to the definition of the local interaction rule given in eq. (7), (at least) these agents will always send the neutral element $\mathfrak{n}$ to their fathers for any further aggregation (Figure 2-a). Now, let us consider a new interval of time $[t_1,\, t_1 + \Delta t_1]$ with $t_1 = t_0 + \Delta t_0 + 1$. We can use the same argument with respect to a new spanning-tree $\mathcal{T}'$ obtained by removing the leaves from the original spanning-tree $\mathcal{T}$. In fact, there are some other agents which play the role of leaves for the new spanning-tree $\mathcal{T}'$ in the time interval $[t_1,\, t_1 + \Delta t_1]$. This implies again that (at least) these agents will always send the neutral element $\mathfrak{n}$ for any further aggregation to their fathers. At this point, since the number of agents is finite, by repeating this reasoning it will exist an interval $[t_h,\, t_h + \Delta t_h]$ after which the residual spanning tree $\mathcal{T}^h$ will be composed of only one agent i, whose state $\bar{s}$ is the aggregation of all the observations available over the network (Figure 2-b). Let us now consider, a new spanning tree $\mathcal{T}^{h+1}$ composed of such an agent i and all of its one-hop neighbors. There will exist an interval $[t_{h+1},\, t_{h+1} + \Delta t_{h+1}]$ after which the forest $\mathcal{F}(t_{h+1}, t_{h+1}+\Delta t_{h+1})$ is connected. As a result, all the agents belonging to this spanning tree will have reached the same knowledge as the agent i. This is due to the fact, that agent i will be the only one to send a novelty different from $\mathfrak{n}$, and therefore any aggregation will let the other agents reach its state $\bar{s}$. By iterating the same reasoning, there will be an interval of time $[t_{2h},\, t_{2h} + \Delta t_{2h}]$ for which the related spanning tree $\mathcal{T}^{2h}$ will coincide with the original spanning tree $\mathcal{T}$. At this point, all the agents will have reached the same state $\bar{s}$ as the agent i (Figure 2-c). Therefore, according to the proof of existence, $\bar{s}$ is a steady state for the multi-agent system.

3) Steady-State Uniqueness: In order to prove the uniqueness of the steady state, it will be shown that any sequence of aggregations over the network, where each agent is considered at least once, is always the combination of the initial set of observations, that is:

$$\bar{s}(t) = s_1(0) \oplus s_2(0) \oplus \ldots \oplus s_n(0).$$

This can be proven by recalling the properties given in Lemma 2. In fact, the particular sequence of aggregations does not affect the result due to the commutativity and associativity properties, while the presence of several occurrences of the same state can be neglected due to the idempotence property. As a result, the combination of the initial set of observations is achieved. At this point, by exploiting the result given in Lemma 3, the following holds:

$$\begin{aligned} \bar{s}(t) &= s_1(0) \oplus s_2(0) \oplus \ldots \oplus s_n(0) \\ &= s_1(0) \otimes s_2(0) \otimes \ldots \otimes s_n(0). \end{aligned}$$

Thus proving the theorem. Details concerning the algorithm execution are provided in Section V. ■

Let us now provide a characterization of the convergence time with respect to a given edge selection process $\mathfrak{e}$.

***Lemma** 4 (Convergence Time):* Let us consider an edge selection process $\mathfrak{e}$ such that $\forall\, t\; \exists \Delta t \in \mathbb{N}$, so that the forest $\mathcal{F}(t, t + \Delta t)$ is connected. If $\exists\, M \in \mathbb{N} \;:\; \Delta t < M\; \forall t$, then the convergence is reached by any agent at most at time $\bar{t} = d \cdot M$, where d is the diameter of the spanning tree $\mathcal{T}$.

Proof: The proof of the lemma follows the same arguments of the steady-state convergence proof (Section IV-2) by assuming that an upper bound is available to the time required for the forest to be connected. In particular, for sake of simplicity and with no lack of generality let us assume to start at time $t = 0$. Under this assumption, the information contraction process towards a single agent i described in Section IV-2 takes in the worst case, i.e., the leaves are the last agents to perform an aggregation, time $t_1 = (d/2) \cdot M$. In the same way, the information propagation process from such an agent i to all the other agents over the network described in Section IV-2 takes in the worst case, i.e., one of the leaves of the previous spanning-tree is the last agent to perform an aggregation, time $t_2 = (d/2) \cdot M$. Therefore, the overall time required to the algorithm to converge in the worst case scenario is $t_{\text{tot}} = t_1 + t_2 = d \cdot M$. ■

As far as the computational complexity of the proposed algorithm is concerned, from the single agent perspective, each time an aggregation is performed the following operations have to be carried out: 1) Novelty extraction, 2) BBAs aggregation. Since, these two operations have the same computational complexity, the first being the inverse of the second, as proven in Lemma (1), the asymptotic computational complexity of the interaction rule $\mathcal{R}$ is of the same order as for the Smets aggregation. Regarding the memory occupancy, each agent should be able to store the common BBA for each of its neighbors. Thus in the worst-case scenario, that is a node linked to each other one, the memory requirement is of the order of the number of agents n.

V. Networked TBM - Static Scenario: An example

In the following, the classification problem introduced in Section III is reviewed. Two possible classes of targets, e.g., “cars” or “trucks” (“a” or “b” in the following), are considered. Hence the following frame of discernment is defined $\Omega = \{a, b\}$ and the following power-set is obtained, $\Gamma = \{\emptyset, a, b, \{a, b\}\}$. The classification task is supposed to be carried out by a system composed of 5 agents. The aggregated sensor readings provided by the agents on the classes are expressed by the BBAs detailed in Table II.

Table III depicts the result of the centralized aggregation schema detailed in Definition 5. In particular, according to the knowledge about the class of targets obtained, the system classifies the target as a truck.

Let us now consider the distributed TBM aggregation schema described in Section IV. Figure 4 depicts the multi-agent system where the solid (black) lines describes the network topology $\mathcal{G}$, while the dashed (red) lines represents the spanning tree $\mathcal{T}$ computed in a distributed fashion by the agents.

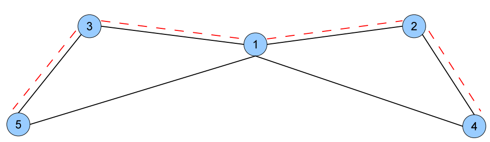


Fig. 4. Multi-Agent system: solid (black) lines represent the network topology $\mathcal{G}$, dashed (red) lines describe the spanning tree $\mathcal{T}$.

Table IV describes the adopted edge sequence as output of the selection process. Note that, only collaborations among agents which augment their common knowledge, i.e., the novelty is not the neutral element $\mathfrak{n}$ at least for one of the two agents, have been considered. This allows the system to converge after only 9 steps which is significantly lower compared to the upper bound provided in Lemma 4 where $d = M = 4$. This can be explained by the fact that while the upper bound has been obtained considering the worst-case scenario, the numerical example presents an optimized edge selection process, i.e., for each collaboration at least one of the two agents augments its knowledge.

Tables XI, XII, XIII show the result of the distributed data aggregation process based upon the local interaction rule given in Theorem 1. In particular, the multi-agent system converges towards the same knowledge about the class of targets as for the centralized aggregation schema. Thus each agent correctly classifies the target as a truck.

Note that, as pointed out in Section III, the straightforward application of the Smets operator would lead to a wrong target classification, i.e., different from the centralized one given in Definition 5, if the edge selection process described in Table IV were considered. This is due to the fact that starting from time $t = 5$ agents are required to collaborate more than once, thus the information brought by these nodes would be erroneously considered several times. This can be viewed by comparing the last two columns of Tables IX, X, XI, XII, XIII, where the result of the aggregation for both the proposed local interaction rule and the Smets operator are given.

Regarding the memory load, by assuming each mass can be represent with a double (4 bytes in our representation), the memory occupancy for each agent is equal to 4 x 2 x 8 = 64 bytes, where 2 is the maximum number of neighbors in the spanning-tree $\mathcal{T}$ and 8 is the cardinality of the power-set Γ.

Set #	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
$\emptyset$	0	0	0	0	0
$\{a\}$	0.1	0.2	0.1	0.2	0.3
$\{b\}$	0.8	0.7	0.8	0.7	0.4
$\{a, b\}$	0.1	0.1	0.1	0.1	0.3

TABLE II
Observations collected by the system of 5 agents.

Set #	Agent 12	Agent 123	Agent 1234	Agent 12345	C-TBM
$\emptyset$	0.23	0.341	0.4781	0.63499	0.63499
$\{a\}$	0.05	0.011	0.0035	0.00213	0.00213
$\{b\}$	0.71	0.647	0.5183	0.36285	0.36285
$\{a, b\}$	0.01	0.001	0.0001	0.00003	0.00003

TABLE III
Centralized TBM: final result and progressive aggregation.

Time	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9
Edge	e_{12}	e_{13}	e_{24}	e_{35}	e_{12}	e_{13}	e_{12}	e_{35}	e_{24}

TABLE IV
Edge selection process.

Set #	Agent 1	$\bar{s}_{1,2}$	$\tilde{s}_{1,2}$	Agent 2	$s_1 \oplus s_2$
$\emptyset$	0	0	0	0	0.23
$\{a\}$	0.1	0	0.1	0.2	0.05
$\{b\}$	0.8	0	0.8	0.7	0.71
$\{a, b\}$	0.1	1	0.1	0.1	0.01

TABLE V
Distributed TBM: t=1, $s_1 \oplus s_2$

Set #	Agent 1	$\bar{s}_{1,3}$	$\tilde{s}_{1,3}$	Agent 3	$s_1 \oplus s_3$
$\emptyset$	0.23	0	0.23	0	0.341
$\{a\}$	0.05	0	0.05	0.1	0.011
$\{b\}$	0.71	0	0.71	0.8	0.647
$\{a, b\}$	0.01	1	0.01	0.1	0.001

TABLE VI
Distributed TBM: t=2, $s_1 \oplus s_3$

Set #	Agent 2	$\bar{s}_{2,4}$	$\widetilde{s}_{2,4}$	Agent 4	$s_2 \oplus s_4$
$\emptyset$	0.23	0	0.23	0	0.407
$\{a\}$	0.05	0	0.05	0.2	0.017
$\{b\}$	0.71	0	0.71	0.7	0.575
$\{a,\, b\}$	0.01	1	0.01	0.1	0.001

TABLE VII
DISTRIBUTED TBM: T=3, $s_2 \oplus s_4$

Set #	Agent 3	$\bar{s}_{3,5}$	$\widetilde{s}_{3,5}$	Agent 5	$s_3 \oplus s_5$
$\emptyset$	0.341	0	0.341	0	0.5395
$\{a\}$	0.011	0	0.011	0.3	0.0069
$\{b\}$	0.647	0	0.647	0.4	0.4533
$\{a,\, b\}$	0.001	1	0.001	0.3	0.0003

TABLE VIII
DISTRIBUTED TBM: T=4, $s_3 \oplus s_5$

Set #	Agent 1	$\bar{s}_{1,2}$	$\widetilde{s}_{1,2}$	Agent 2	$s_1 \oplus s_2$	$s_1 \otimes s_2$
$\emptyset$	0.341	0.23	0	0.407	0.4781	0.626537
$\{a\}$	0.011	0.05	0.1	0.017	0.0035	0.000215
$\{b\}$	0.647	0.71	0.8	0.575	0.5183	0.373247
$\{a,\, b\}$	0.001	0.01	0.1	0.001	0.0001	0.000001

TABLE IX
DISTRIBUTED TBM: T=5, $s_1 \oplus s_2$

Set #	Agent 1	$\bar{s}_{1,3}$	$\widetilde{s}_{1,3}$	Agent 3	$s_1 \oplus s_3$	$s_1 \otimes s_3$
$\emptyset$	0.4781	0.341	0	0.5395	0.63499	0.7385971
$\{a\}$	0.0035	0.011	0.2	0.0069	0.00213	0.0001293
$\{b\}$	0.5183	0.647	0.7	0.4533	0.36285	0.2612733
$\{a,\, b\}$	0.0001	0.001	0.1	0.0003	0.00003	0.0000003

TABLE X
DISTRIBUTED TBM: T=6, $s_1 \oplus s_3$

Set #	Agent 1	$\bar{s}_{1,2}$	$\widetilde{s}_{1,2}$	Agent 2	$s_1 \oplus s_2$	$s_1 \otimes s_2$
$\emptyset$	0.63499	0.4781	0	0.4781	0.63499	0.84950
$\{a\}$	0.00213	0.0035	0.3	0.0035	0.00213	$2.3 \cdot 10^{-6}$
$\{b\}$	0.36285	0.5183	0.4	0.5183	0.36285	0.15049
$\{a,\, b\}$	0.00003	0.0001	0.3	0.0001	0.00003	$3 \cdot 10^{-10}$

TABLE XI
DISTRIBUTED TBM: T=7, $s_1 \oplus s_2$

Set #	Agent 3	$\bar{s}_{3,5}$	$\widetilde{s}_{3,5}$	Agent 5	$s_3 \oplus s_5$	$s_3 \otimes s_5$
$\emptyset$	0.63499	0.5395	0	0.5395	0.63499	0.931736
$\{a\}$	0.00213	0.0069	0.2	0.0069	0.00213	$1.6 \cdot 10^{-8}$
$\{b\}$	0.36285	0.4533	0.7	0.4533	0.36285	0.068263
$\{a,\, b\}$	0.00003	0.0003	0.1	0.0003	0.00003	$9 \cdot 10^{-14}$

TABLE XII
DISTRIBUTED TBM: T=8, $s_3 \oplus s_5$

Set #	Agent 2	$\bar{s}_{2,4}$	$\widetilde{s}_{2,4}$	Agent 4	$s_2 \oplus s_4$	$s_2 \otimes s_4$
$\emptyset$	0.63499	0.407	0.28	0.407	0.63499	0.913315
$\{a\}$	0.00213	0.017	0.09	0.017	0.00213	$4.2 \cdot 10^{-8}$
$\{b\}$	0.36285	0.575	0.6	0.575	0.36285	0.086684
$\{a,\, b\}$	0.00003	0.001	0.03	0.001	0.00003	$3 \cdot 10^{-13}$

TABLE XIII
DISTRIBUTED TBM: T=9, $s_2 \oplus s_4$

VI. DISTRIBUTED DATA AGGREGATION VIA NETWORKED TBM - DYNAMIC SCENARIO

In this section a local interaction rule $\mathcal{R}'$ to perform distributed TBM data aggregation, in case of dynamic observations, is described. The following assumption on the storage on the storing capabilities of an agent are made:

***Assumptions** 2:*
In this framework, each agent i can handle the storing of the following information up to time t:

- $s_i(t)$ current knowledge;
- $\bar{s}_{i,j}(t)$ common knowledge with agent j so that $e_{ij} \in \hat{E}$;
- $p_i(t)$ BBA related to the previous observation;
- $l_i(t)$ BBA related to the latest observation;

For sake of clarity, the following notation will be used indiscriminately in the rest of the paper:

$$s_c(t) = \bar{s}_{i,j}(t) \qquad s_s(t) = s_i(t) \otimes s_j(t)$$

***Definition** 7 (Combination rules):* In a dynamic scenario each agent can perform two type of data aggregation:

- Local Aggregation performed by a single agent.
- Dynamic Aggregation performed by a couple of agents.

A. Local Aggregation

An agent requires a local aggregation any time a new observation is available. To this end, let us suppose that the initial observation is made at time $t = 0$ and a new one is collected at time $\hat{t}_k,\ \ k \in \mathbb{N}$. In this context, an agent must perform a *local aggregation* in order to remove the past observation from its current knowledge and add the new one. This can be achieved by first computing the novelty between the current knowledge s_i and the past observation p_i and then aggregating this novelty $\tilde{s}_i^p$ with the latest observation l_i. In this way, the updated current knowledge of the agent will take into account only the information coming from the previous collaboration with its neighbors and its last observation, that is:

$$s_i(\hat{t}_k) = \tilde{s}_i^p(\hat{t}_k) \otimes l_i(\hat{t}_k) \tag{14}$$

B. Dynamic Aggregation Between Agents

In order to perform the dynamic aggregation between two agents the operator $\vec{\oplus}$ must be introduced.

***Definition** 8 ($\mathcal{R}'$ - operator $\vec{\oplus}$):*
Let $\mathcal{R}'$ be a rule to combine the basic belief assignments for two agents (i, j) such that $e_{ij} \in \hat{E}$ as follows:

$$\begin{aligned} s_i(t+1) &= s_i(t) \,\vec{\oplus}\, s_j(t) = \widetilde{(s_i(t) \otimes s_j(t))}^{c} \\ &= \Big\{ \tilde{m}_s^c(t, \gamma_a) \Big\}, \quad \forall \gamma_a \in \Gamma \end{aligned} \tag{15}$$

where $\tilde{m}_s^c(t, \gamma_a)$ is defined as in eq. (6) with respect to $s_s(t)$ and $s_c(t)$.

***Remark** 2:* A few remarks are now in order:
1) A simple consequence of Lemma 1 with respect to the operator $\vec{\oplus}$ is that the aggregation of two state s_i and s_j such that $s_j = \bar{s}_{i,j}$ is:

$$s_j(t+1) = \widetilde{(s_j(t) \otimes s_i(t))}^{c} = \widetilde{(s_j(t) \otimes s_i(t))}^{j} = s_i(t)$$

2) According to the operator $\vec{\oplus}$, two states $s_i(t)$ and $s_j(t)$ are equal if and only if they are completely described by their common knowledge, i.e. $s_i(t) = s_j(t) = \bar{s}_{i,j}(t) = s_c(t)$.

Therefore, the aggregation rule used for the dynamic scenario turns out to be the same as in the static scenario if the agents do not collect new observations over time.

At this point, in order to prove the convergence of the proposed algorithm some properties concerning the local interaction rule $\mathcal{R}'$ must be introduced.

***Lemma** 5 ($\vec{\bigoplus}$ properties):* The local interaction rule $\vec{\bigoplus}$ defined according to eq. (7) has the following properties:

$$
\begin{array}{rcll}
s_i \vec{\bigoplus} s_j & = & s_j \vec{\bigoplus} s_i & \text{(commutativity)} \\
s_i \vec{\bigoplus} s_i & = & s_i \quad \text{if } \bar{s}_{i,j} = s_i & \text{(idempotence)} \\
(s_i \vec{\bigoplus} s_j) \vec{\bigoplus} s_k & = & s_i \vec{\bigoplus} (s_j \vec{\bigoplus} s_k) & \text{(associativity)}
\end{array}
\tag{16}
$$

for each triple $(i,\, j,\, k)$ such that $e_{ij}, e_{jk} \in \hat{E}$.

Proof: The properties can be proven by exploiting the Lemma 1 and the definition given in eq. (7). Furthermore, let us consider the more general case for which agents might have already performed an aggregation with each other. In particular, according to Lemma 1, the state s_i of an agent i with respect to any of its neighbors j can be always written as:

$$
\begin{aligned}
s_i(t_m) &= l_i(t_m) \otimes s_{\mathcal{N}(i)\setminus j}(t_m) \otimes l_j(t_q) \otimes s_{\mathcal{N}(j)\setminus i}(t_q) \\
&= v_i^j(t_m) \otimes v_j^i(t_q)
\end{aligned}
$$

where $s_{\mathcal{N}(i)\setminus j}(t_m), s_{\mathcal{N}(j)\setminus i}(t_q)$ describe respectively all the aggregated data coming from the neighborhood of agent i (at time t_m) and j (at time t_q) excluding each other, and $v_h^p(t_k) = l_h(t_k) \otimes s_{\mathcal{N}(h)\setminus p}(t_k)$.

- Commutativity:
  Let us consider two agents (i, j), then from Definition 8 we have:
  $$s_i \vec{\bigoplus} s_j = (\widetilde{s_i \otimes s_j})^{c} = (\widetilde{s_j \otimes s_i})^{c} = s_j \vec{\bigoplus} s_i$$
  where $\big(s_i \otimes s_j\big) = \big(s_j \otimes s_i\big)$ comes from the commutativity property of the Smets operator $\otimes$.
- Idempotence:
  Let us consider two agents (i, j) that a given time t have their BBA equal to their common knowledge (acquired at a certain instant of time previous the time t), that is $s_i = s_j = \bar{s}_{i,j}$, then we have
  $$s_i \vec{\bigoplus} s_j = (\widetilde{s_i \otimes s_j})^{c} = (\widetilde{s_i \otimes s_j})^{j} = s_i$$
- Associativity:
  Let us consider a triplet of agents $(i,\, j,\, k)$ such that $e_{ij}, e_{jk} \in \hat{E}$. Furthermore, according to Lemma 1, let us assume the current state of the three agents at time t_m to be written as follows:
  $$
  \begin{aligned}
  s_i &= v_i^j(t_m) \otimes v_j^i(t_q) \\
  s_j &= v_i^j(t_q) \otimes l_j(t_m) \otimes s_{\mathcal{N}(j)\setminus i,k}(t_m) \otimes v_k^j(t_{q'}) \\
  &= v_i^j(t_q) \otimes v_j^{i,k}(t_m) \otimes v_k^j(t_{q'}) \\
  s_k &= v_k^j(t_m) \otimes v_j^k(t_{q'})
  \end{aligned}
  $$
  and the common knowledge describing previous aggregation among these agents to be written as:
  $$
  \begin{aligned}
  \bar{s}_{i,j}(t_q) &= s_{c'} = & v_i^j(t_q) \otimes v_j^i(t_q) \\
  \bar{s}_{j,k}(t_{q'}) &= s_{c''} = & v_j^k(t_{q'}) \otimes v_k^j(t_{q'}).
  \end{aligned}
  $$
  Then we have:
  $$
  \begin{aligned}
  &\Big(s_i(t_m) \vec{\bigoplus} s_j(t_m)\Big) \vec{\bigoplus} s_k(t_m) = \\
  &= \Big(v_i^j(t_m) \otimes v_j^i(t_q) \otimes \widetilde{v_i^j(t_q)} \otimes v_j^{i,k}(t_m) \otimes v_k^j(t_{q'})\Big)^{c'} \vec{\bigoplus} s_k(t_m) \\
  &= \Big(v_i^j(t_m) \otimes v_j^{i,k}(t_m) \otimes v_k^j(t_{q'})\Big) \vec{\bigoplus} s_k(t_m) \\
  &= s_{ij}(t_m) \vec{\bigoplus} s_k(t_m) \\
  &= \Big(v_i^j(t_m) \otimes v_j^{i,k}\widetilde{(t_m) \otimes} v_k^j(t_{q'}) \otimes s_k(t_m)\Big)^{c''} \\
  &= \Big(v_i^j(t_m) \otimes v_j^{i,k}(t_m) \otimes \widetilde{v_k^j(t_{q'})} \otimes v_k^j(t_m) \otimes v_j^k(t_{q'})\Big)^{c''} \\
  &= v_i^j(t_m) \otimes v_j^{i,k}(t_m) \otimes v_k^j(t_m)
  \end{aligned}
  $$
  where the equivalence $\bar{s}_{ij,k} = \bar{s}_{j,k}$ comes from the independence of the knowledge between node i and k with respect to j, due to the properties of the topology structure of the communication graph, i.e., a spanning tree $\mathcal{T}$. And:
  $$
  \begin{aligned}
  &s_i(t_m) \vec{\bigoplus} \Big(s_j(t_m) \vec{\bigoplus} s_k(t_m)\Big) = \\
  &= s_i(t_m) \vec{\bigoplus} \Big(v_i^j(t_q) \otimes v_j^{i,k}(t_m) \otimes \widetilde{v_k^j(t_{q'})} \otimes v_k^j(t_m) \otimes v_j^k(t_{q'})\Big)^{c''} \\
  &= s_i(t_m) \vec{\bigoplus} \Big(v_i^j(t_q) \otimes v_j^{i,k}(t_m) \otimes v_k^j(t_m)\Big) \\
  &= s_i(t_m) \vec{\bigoplus} s_{jk}(t_m) \\
  &= \Big(s_i(t_m) \otimes v_i^j\widetilde{(t_q) \otimes v}_j^{i,k}(t_m) \otimes v_k^j(t_m)\Big)^{c'} \\
  &= \Big(v_i^j(t_m) \otimes v_j^i(t_q) \otimes \widetilde{v_i^j(t_q)} \otimes v_j^{i,k}(t_m) \otimes v_k^j(t_m)\Big)^{c'} \\
  &= v_i^j(t_m) \otimes v_j^{i,k}(t_m) \otimes v_k^j(t_m)
  \end{aligned}
  $$
  where the equivalence $\bar{s}_{i,jk} = \bar{s}_{i,j}$ comes again from the independence of the knowledge between node i and k with respect to j, due to the properties of the topology structure of the communication graph, i.e., a spanning tree $\mathcal{T}$.

■

In a dynamic scenario, referring to Definition (5), let us assume that every so often ($t = \hat{t}_k,\;\; k \in \mathbb{N}$), one or more agents perform an update of their observation. As a consequence, the aggregated BBA can be updated accordingly as follows:

$$s_{1\,2\,\ldots\,n}(\hat{t}_k) = l_1(\hat{t}_k) \otimes l_2(\hat{t}_k) \otimes \ldots \otimes l_n(\hat{t}_k) \tag{17}$$

where $l_i(\hat{t}_k)$ describes the BBA related to the most recent observation available to the agent i.

For the dynamic scenario, the convergence of the proposed gossip algorithm towards the basic belief assignment, as in the centralized aggregation schema, is guaranteed by the following theorem.

***Theorem** 2 (Distributed Dynamic TBM):* Let us consider a gossip algorithm $\{\mathcal{S}, \mathcal{R}', \mathfrak{e}\}$ over a spanning-tree $\mathcal{T} = \{V, \hat{E}\}$ with $\mathcal{S}$ and $\mathcal{R}'$ defined respectively as in Definition 3 and Definition 8. Let us assume each agent i at time t$=\;0$ provides an independent observation described by the basic belief assignment $s_i(0) = \{m_i(0, \gamma_a\},\; \gamma_a \in \Gamma\}$. Furthermore, let us assume that every so often ($t = \hat{t}_k,\; k \in \mathbb{N}$), one or more agents perform an update of their observation. If $\mathfrak{e}$ is such that $\forall\, t \quad \exists\, \Delta t \in \mathbb{N}$ so that the time-variant forest $\mathcal{F}(t, t+\Delta t)$ is connected, then for some k there will exist a time $t = \bar{t}_k$ so that:

$$s_i(t') = l_1(\hat{t}_k) \otimes l_2(\hat{t}_k) \otimes \ldots \otimes l_n(\hat{t}_k)\ \forall\, t' \in [\,\bar{t}_k,\ \hat{t}_{k+1}\,) \quad (18)$$

where $l_i(\hat{t}_k)$ describes the most recent observation available to the agent i.

Proof: The proof of the theorem consists of four steps. First, it will be proven that a steady-state exists for the proposed gossip algorithm. Successively, it will be shown that each agent by applying the aggregation operator $\vec{\oplus}$ can inject the updated observations into the network while transparently removing the previous ones. Then, by exploiting this property, it will be proven the convergence and finally, it will be shown that the steady-state is unique and equal to the result of the centralized aggregation schema given in eq. (17).

Note that, for the existence, convergence and uniqueness analysis, the interval of time $[\hat{t}_k,\, \hat{t}_{k+1})$ between two consecutive observations update is supposed to be long enough with respect to the nature of the edge selection process $\mathfrak{e}$. This allows to guarantee that each agent can perform all the aggregations required to reach the steady-state. Necessary and sufficient conditions concerning the length of time interval with respect to the edge selection process $\mathfrak{e}$ are given in Lemma 6.

1) Steady-State Existence: In order to prove the existence of a steady-state for the proposed gossip algorithm, it will be shown that a sufficient and necessary condition is that all the agents share the same state $\bar{s}$. According to the interaction rule given in eq. (7), this imply that all the agents will have the same common knowledge s_c. Therefore, the state $\bar{s}$ is itself a steady state for the multi-agent system. Furthermore, let us prove by contradiction this condition to be necessary as well. To this end, let us suppose two agents i and j have reached two different steady states over the network, that is $s_i(t) = s'$ and $s_j(t) = s''$. Therefore, according to the definition of a spanning-tree, there will always exist a (unique) path connecting the two nodes i and j. Let us now consider for such a spanning-tree $\mathcal{T}$ the path $p_{ij} = \{v_i,\, v_{k\in\mathcal{N}_i},\, \ldots,\, v_{h\in\mathcal{N}_j},\, v_j\}$ connecting these two agents i and j. In particular, as agent i has reached the state s', this implies that its neighbor k must be sending a state which is equal to their common knowledge. Furthermore, since agent k itself has reached a steady-state, agent i must be sending a state which is equal to their common knowledge. However, according to the Remark 2 this implies that both agents have the same state, that is s'. The same argument can be applied to the agent j and its neighbor h with respect to the steady state s''.

Now, by iterating this reasoning from both ends of the path there will be a cut where all the nodes on a side will have reached the same steady-state s' as agent i, while on the other side all the agents will have reached the same steady-state s'' as agent j. Let us call x and y the two agents on the boundaries of the cut. Since, x and y have boot reached a steady state, s' and s'' respectively, they must be sending a state which is equal to their common knowledge to each other. However, the two steady states s' and s'' have been supposed to be different, so the two agents x and y *cannot* be sending a state equal to their common knowledge s_c to each other. Indeed, this would be possible only if the two steady states s' and s'' were the same steady state $\bar{s}$, which gives the absurd. Therefore a steady state holds if and only if $s_i = s_j \quad \forall i, j \in N$.

2) Observations Propagation: Let us consider two agents (i, j) such that $e_{ij} \in \hat{E}$ and let us assume a collaboration was performed at time $t = t_q$. The updated states can be written as:

$$\begin{aligned}
s_i(t_q) &= l_i(t_q) \otimes s_{\mathcal{N}(i)\setminus j}(t_q) \otimes l_j(t_q) \otimes s_{\mathcal{N}(j)\setminus i}(t_q)\\
&= v_i^j(t_q) \otimes v_j^i(t_q)\\
s_j(t_q) &= l_j(t_q) \otimes s_{\mathcal{N}(j)\setminus i}(t_q) \otimes l_i(t_q) \otimes s_{\mathcal{N}(i)\setminus q}(t_q)\\
&= v_j^i(t_q) \otimes v_i^j(t_q),
\end{aligned}$$

where $s_{\mathcal{N}(i)\setminus j}(t_q)$, $s_{\mathcal{N}(j)\setminus i}(t_q)$ describe respectively all the aggregated data coming from the neighborhood of agent i and j excluding each other, and $v_h^p(t_k) = l_h(t_k) \otimes s_{\mathcal{N}(h)\setminus p}(t_k)$. Furthermore the common knowledge at time $t = t_q$ between the two agents can be written as $s_c(t_q) = v_j^i(t_q) \otimes v_i^j(t_q)$, Now, let us consider a time $t = t_m,\ t_m$ such that the two agents have performed further aggregations (but not with each other) and an update of their observation. Their current state at time $t = t_m$ can be written as follows:

$$\begin{aligned}
s_i(t_m) &= l_i(t_m) \otimes s_{\mathcal{N}(i)\setminus j}(t_m) \otimes l_j(t_q) \otimes s_{\mathcal{N}(j)\setminus i}(t_q)\\
&= v_i^j(t_m) \otimes v_j^i(t_q)\\
s_j(t_m) &= l_j(t_m) \otimes s_{\mathcal{N}(j)\setminus i}(t_m) \otimes l_i(t_q) \otimes s_{\mathcal{N}(i)\setminus q}(t_q)\\
&= v_j^i(t_m) \otimes v_i^j(t_q).
\end{aligned}$$

Now, let us assume that the two agents perform an aggregation at time $t = t_m + 1$. Their current state can be updated accordingly as follows:

$$\begin{aligned}
s_i(t_m+1) &= s_i(t_m) \vec{\otimes} s_j(t_m)\\
&= \Big(v_i^j(t_m) \otimes \overbrace{v_j^i(t_q) \otimes v_j^i(t_m)}\otimes v_i^j(t_q)\Big)^c\\
&= v_i^j(t_m) \otimes v_j^i(t_m)\\
s_j(t_m+1) &= s_j(t_m) \vec{\otimes} s_i(t_m)\\
&= \Big(v_j^i(t_m) \otimes \overbrace{v_i^j(t_q) \otimes v_i^j(t_m)}\otimes v_j^i(t_q)\Big)^c\\
&= v_j^i(t_m) \otimes v_i^j(t_m)
\end{aligned}$$

where the common knowledge $s_c(t_q) = v_j^i(t_q) \otimes v_i^j(t_q)$, which represents both their previous observations and their neighbors previous observations, is removed. Note that, after the aggregation the common knowledge is set to the current state of the two agents $s_c(t_m + 1) = v_i^j(t_m) \otimes v_j^i(t_m)$. Therefore, anytime two agents perform an aggregation only the most recent observation of any agent is propagated over the network.

3) Steady-State Convergence: In order to prove the convergence of the proposed algorithm towards a steady-state, let us consider an interval of time $[t_0, t_0 + \Delta t_0]$, with $t_0 = \hat{t}_k$, for which the forest $\mathcal{F}(t_0, t_0+\Delta t_0)$ is connected. This implies that some agents play the role of leaves for the resulting spanning-tree $\mathcal{T}$. Indeed, any further aggregation of these agents with their fathers will not change the state of the fathers. This is due to the fact that all the knowledge brought by the leaves is already available to the fathers in their common knowledge (see Remark 2). Furthermore, according to the Observation Propagation proof (Section VI-B2) only the most recent observations will be sent by the leaves to the fathers. Now, let us consider a new interval of time $[t_1,\, t_1 + \Delta t_1]$ with $t_1 = t_0 + \Delta t_0 + 1$. We can use the same argument with respect to a new spanning-tree $\mathcal{T}'$ obtained by removing the leaves from the original spanning-tree $\mathcal{T}$. In fact, there are some other agents which play the role of leaves for the new spanning-tree $\mathcal{T}'$ in the time interval $[t_1,\, t_1 + \Delta t_1]$. This implies again that (at least) these agents will always send the common knowledge s'_c for any further aggregation to their fathers. At this point, since the number of agents is finite, by repeating this reasoning it will exist an interval $[t_h,\, t_h + \Delta t_h]$ after which the residual spanning tree $\mathcal{T}^h$ will be composed of only one agent i, whose state $\bar{s}$ is the aggregation of all the *most recent observations* available (at time $\hat{t}_k$) over the network. Let us now consider, a new spanning tree $\mathcal{T}^{h+1}$ composed of such an agent i and all of its one-hop neighbors. There will exist an interval $[t_{h+1},\, t_{h+1} + \Delta t_{h+1}]$ after which the forest $\mathcal{F}(t_{h+1}, t_{h+1} + \Delta t_{h+1})$ is connected. As a result, all the agents belonging to this spanning tree will have reached the same knowledge as the agent i. This is due to the fact that for any aggregation, agent i will be the only to have its state different from the common knowledge. Therefore according to the Remark (2), the other agents will reach its state $\bar{s}$. By iterating the same reasoning, there will be an interval of time $[t_{2h},\, t_{2h} + \Delta t_{2h}]$ for which the related spanning tree $\mathcal{T}^{2h}$ will coincide with the original spanning tree $\mathcal{T}$. This implies that all the agents will be reached the same state $\bar{s}$ as the agent i. Therefore, according to the proof of existence, $\bar{s}$ is a steady state for the multi-agent system.

4) Steady-State Uniqueness: In order to prove the uniqueness of the steady state, it will be shown that any sequence of aggregations over the network, where each agent is considered at least once, is always the $\vec{\oplus}$ combination of the observations set at time $t = \hat{t}_k$, that is:

$$\bar{s}(t') = s_1(\hat{t}_k) \,\vec{\bigoplus}\, s_2(\hat{t}_k) \,\vec{\bigoplus}\, \ldots \,\vec{\bigoplus}\, s_n(\hat{t}_k), \quad \forall\, t' \in [\bar{t}_k,\, \hat{t}_{k+1}).$$

This can be proven by recalling the properties given in Lemma 5 and the result concerning the proof of the Observation Propagation (Section VI-B2) along with the proof of Steady-State Convergence (Section VI-B3). In fact, the observation propagation result guarantees that, anytime two agents perform an aggregation, only the most recent observation of any agent is propagated over the network. Furthermore, according to the convergence proof, when a steady-state is reached over the network, it embodies all the most recent observations available up to time t_k. Finally, due to Lemma **??**, the particular sequence of aggregations does not affect the result due to the commutativity and associativity properties, while the presence of several occurrences of the same state can be neglected due to the idempotence property. As a result, the combination of the observations set at time $t = \hat{t}_k$ is achieved, that is:

$$\bar{s}(t) = l_1(\hat{t}_k) \,\vec{\bigoplus}\, l_2(\hat{t}_k) \,\vec{\bigoplus}\, \ldots \,\vec{\bigoplus}\, l_n(\hat{t}_k),$$

that is the same result as in the centralized aggregation schema given in eq. (17). Details concerning the algorithm execution are provided in Section VII. ■

***Remark** 3:* A few important remarks are now in order:

1) The gossip algorithm described in the dynamic case allows the agents to "track" the steady-state (given by eq. (17)). In fact, by applying the local aggregation rule given in eq. (14) each agent can replace the previous observation with the latest one on its own state, and by applying the dynamic aggregation rule given in eq. (7) if two agents perform an aggregation only the most recent observation of any agent is propagated over the network.
2) The convergence capability of the proposed algorithm depends on whether the time interval between two consecutive observations update is sufficiently long with respect to the nature of the edge selection process $\mathfrak{e}$. However, even if some steady-states are missed, the agents still keep tracking the most recent one.

In the following, an analysis to derive an upper-bound of the convergence time for the worst-case scenario is proposed.

***Lemma** 6 (Convergence Time):* Let us consider an edge selection process $\mathfrak{e}$ such that $\forall\, t\; \exists\, \Delta t \in \mathbb{N}$ so that the forest $\mathcal{F}(t, t{+}\Delta t)$ is connected. If $\exists\, M \in \mathbb{N} \;:\; \Delta t < M \;\forall\, t$, then the multi-agent system can always reach the convergence towards a steady-state if the following condition holds between two consecutive observations update:

$$\hat{t}_{k+1} \geq \hat{t}_k + d \cdot M, \quad \forall\, k \in \mathbb{N} \tag{19}$$

where d is the diameter of the spanning tree $\mathcal{T}$.

Proof: The proof follows the same argument of the steady-state convergence proof (Section VI-B3) by assuming that an upper bound is available to the time required for the forest to be connected. Furthermore, let us assume that at time $t = \hat{t}_k$, one or more agents have performed an observation update over the network. Under this assumption, the information contraction process towards a single agent i described in Section VI-B3 takes in the worst case, i.e., the leaves are the last agents to perform an aggregation, time $t_1 = (d/2) \cdot M$. In particular, the state of such an agent i represents the aggregation of the latest set of observation available over the network up to time t_k. In the same way, the information propagation process from such an agent i to all the other agents over the network described in Section VI-B3 takes in the worst case, i.e., one of the leaves of the previous spanning-tree is the last agent to perform an aggregation, time $t_2 = (d/2) \cdot M$. Therefore, the overall time required to the algorithm to converge in the worst case scenario is $t_{\text{tot}} = t_1 + t_2 = d \cdot M$. Note that, in the case an update is performed by any agent before the contraction process ends, i.e., $\hat{t}_{k+1} < \hat{t}_k + d \cdot M$, the state spread by agent i will no longer represent the aggregation of the most recent set of observations available over the network, and therefore at the end of the propagation process, no steady-state will be reached for the interval $[\hat{t}_k, t_{k+1})$. ■

Note that Lemma 6 provides only a theoretical characterization of the convergence time for the proposed gossip algorithm. However, in a real scenario agents perform the update of their observations independently and asynchronously, therefore no control can be provided for the convergence of the algorithm apart from the design of a "smart" edge selection process able to keep the upper-bound M as small as possible.

As far as the computational complexity of the proposed algorithm for the dynamic scenario is concerned, it should be noticed that from the single agent perspective, the same operations as for the static scenario have to be carried out although with an inverse order: 1) BBAs aggregation, 2) Novelty extraction. Therefore, the same considerations hold. Regarding the memory occupancy, two additional BBAs, namely $p_i(t)$ and $l_i(t)$ have to be stored by each agent, compared to the static scenario. Therefore, also in this case, in the worst-case scenario the memory requirement is of the order of the number of agents n.

VII. Networked TBM - Dynamic Scenario: An Example

In the following, an extension of the example given in Section V where agents collect new observations over the time is proposed. The aggregated sensor readings on the classes of target provided by the agents are expressed by the BBAs given in Table XIV. Note that, an observation update is performed by agents 3 and 4 at time $t = 5$. As a consequence, according to eq. (17) two different aggregated knowledge are available for the centralized aggregation schema, namely one up to time $t = 5$ and the other one for $t \geq 5$. In particular, Table XV shows the aggregated knowledge about the class of targets for $t \geq 5$. According to it, the system classifies the target as a truck.

Set #	$l_1(0)$	$l_2(0)$	$l_3(0)$	$l_4(0)$	$l_5(0)$	$l_3(5)$	$l_4(5)$
$\emptyset$	0	0	0	0	0	0	0
$\{a\}$	0.1	0.2	0.1	0.2	0.3	0.1	0.1
$\{b\}$	0.8	0.7	0.8	0.7	0.4	0.7	0.8
$\{a, b\}$	0.1	0.1	0.1	0.1	0.3	0.2	0.1

TABLE XIV
Observations collected by the system of 5 agents.

Set #	s_{12}	s_{123}	s_{1234}	s_{12345}	CTBM
$\emptyset$	0.23	0.336	0.4134	0.58966	0.58966
$\{a\}$	0.05	0.016	0.0034	0.0021	0.0021
$\{b\}$	0.71	0.646	0.583	0.40818	0.40818
$\{a, b\}$	0.01	0.002	0.0002	0.00006	0.00006

TABLE XV
Centralized TBM: final result and progressive aggregation, using the new observation for agent 3 and 4.

The data aggregation is carried out over the spanning tree $\mathcal{T} = \{V, \hat{E}\}$ with $\hat{E} = \{e_{12}, e_{13}, e_{35}, e_{24}\}$. In particular, Table XVI depicts the set of selected edges. Note that, the convergence cannot be reached by the multi-agent system for the initial set of observations (up to time $t = 0$) due to the update of the agents observations at time $t = 5$. Furthermore, these updates prevent also the application of the local update rule described in Theorem 1 since it does not allow to remove the previous observations from the current knowledge of the agents. Nevertheless, as explained in Section VI, the local update rule given in Theorem 2 can be used instead.

Note that, by applying the local update rule given in Theorem 2, the agents can track the current steady-state regardless of the observations update, hence as pointed out in Remark 3 no re-initialization is required for the system. As a matter of fact, all the agents reaches the steady-state regardless of the update performed by agents 3 and 4. In particular, agents 1 and 2 reach the steady-state at time $t = 9$, as shown in Table XXVI. Obviously, further communications up to time $t = 12$ are required to propagate this result over the network, as shown in Tables XXVII, XXVIII and XXIX.

Set #	s_1	s_2	$s_1 \bigotimes s_2$	$\bar{s}_{1,2}$	$s_1,\ s_2$
$\emptyset$	0	0	0.23	0	0.23
$\{a\}$	0.1	0.2	0.05	0	0.05
$\{b\}$	0.8	0.7	0.71	0	0.71
$\{a, b\}$	0.1	0.1	0.01	1	0.01

TABLE XVII
Dynamic Distributed TBM: t=1, $s_1 \vec{\bigoplus} s_2$

Set #	s_1	s_3	$s_1 \bigotimes s_3$	$\bar{s}_{1,3}$	$s_1,\ s_3$
$\emptyset$	0.23	0	0.341	0	0.341
$\{a\}$	0.05	0.1	0.011	0	0.011
$\{b\}$	0.71	0.8	0.647	0	0.647
$\{a, b\}$	0.01	0.1	0.001	1	0.001

TABLE XVIII
Dynamic Distributed TBM: t=2, $s_1 \vec{\bigoplus} s_3$

Finally, according to Theorem 2 the multi-agent system converges towards the same knowledge about the class of targets as for the centralized aggregation schema. Thus each agent correctly classifies the target as a truck. In addition, it should be noticed how the mass of the empty-set $m(\emptyset)$ has a high value to underline a contradiction of the initial agents observations, as detailed in Table XIV.

Set #	s_2	s_4	$s_2 \bigotimes s_4$	$\bar{s}_{2,4}$	$s_2,\ s_4$
$\emptyset$	0.23	0	0.407	0	0.407
$\{a\}$	0.05	0.2	0.017	0	0.017
$\{b\}$	0.71	0.7	0.575	0	0.575
$\{a, b\}$	0.01	0.1	0.001	1	0.001

TABLE XIX
Dynamic Distributed TBM: t=3, $s_2 \vec{\bigoplus} s_4$

Set #	s_3	s_5	$s_3 \bigotimes s_5$	$\bar{s}_{3,5}$	$s_3,\ s_5$
$\emptyset$	0.341	0	0.5395	0	0.5395
$\{a\}$	0.011	0.3	0.0069	0	0.0069
$\{b\}$	0.647	0.4	0.4533	0	0.4533
$\{a, b\}$	0.001	0.3	0.0003	1	0.0003

TABLE XX
Dynamic Distributed TBM: t=4, $s_3 \vec{\bigoplus} s_5$

VIII. Conclusions

In this work an extension of the Transferable Belief Model to a distributed multi-agent context has been presented. Two different scenarios, namely static scenario and dynamic scenario, have been considered. A distributed protocol has been

Time	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10	t=11	t=12
Edge	e_{12}	e_{13}	e_{24}	e_{35}	u.o.	e_{24}	e_{35}	e_{13}	e_{12}	e_{24}	e_{13}	e_{35}

TABLE XVI
EDGE SELECTION PROCESS. UPDATED OBSERVATIONS AT T=5 (U.O.)

Set #	s_3	p_3	$\tilde{s}_3^{p_3}$	l_3	s_3
$\emptyset$	0.5395	0	0.463	0	0.5362
$\{a\}$	0.0069	0.1	0.033	0.1	0.0102
$\{b\}$	0.4533	0.8	0.501	0.7	0.453
$\{a, b\}$	0.0003	0.1	0.003	0.2	0.0006

TABLE XXI
DYNAMIC DISTRIBUTED TBM: T=5, LOCAL AGGREGATION ON AGENT 3'S KNOWLEDGE.

Set #	s_4	p_4	$\tilde{s}_4^{p_4}$	l_4	s_4
$\emptyset$	0.407	0	0.23	0	0.341
$\{a\}$	0.017	0.2	0.05	0.1	0.011
$\{b\}$	0.575	0.7	0.71	0.8	0.647
$\{a, b\}$	0.001	0.1	0.01	0.1	0.001

TABLE XXII
DYNAMIC DISTRIBUTED TBM: T=5, LOCAL AGGREGATION ON AGENT 4'S KNOWLEDGE.

Set #	s_2	s_4	$s_2 \bigotimes s_4$	$\bar{s}_{2,4}$	$s_2,\ s_4$
$\emptyset$	0.407	0.341	0.626537	0.407	0.341
$\{a\}$	0.017	0.011	0.000215	0.017	0.011
$\{b\}$	0.575	0.647	0.373247	0.575	0.647
$\{a, b\}$	0.001	0.001	0.000001	0.001	0.001

TABLE XXIII
DYNAMIC DISTRIBUTED TBM: T=6, $s_2 \overrightarrow{\bigoplus} s_4$

Set #	s_3	s_5	$s_3 \bigotimes s_5$	$\bar{s}_{3,5}$	$s_3,\ s_5$
$\emptyset$	0.5362	0.5395	0.79416946	0.5395	0.5362
$\{a\}$	0.0102	0.0069	0.00007758	0.0069	0.0102
$\{b\}$	0.453	0.4533	0.20575278	0.4533	0.453
$\{a, b\}$	0.0006	0.0003	0.00000018	0.0003	0.0006

TABLE XXIV
DYNAMIC DISTRIBUTED TBM: T=7, $s_3 \overrightarrow{\bigoplus} s_5$

Set #	s_1	s_3	$s_1 \bigotimes s_3$	$\bar{s}_{1,3}$	$s_1,\ s_3$
$\emptyset$	0.341	0.5362	0.7059382	0.341	0.5362
$\{a\}$	0.011	0.0102	0.000129	0.011	0.0102
$\{b\}$	0.647	0.453	0.2939322	0.647	0.453
$\{a, b\}$	0.001	0.0006	0.0000006	0.001	0.0006

TABLE XXV
DYNAMIC DISTRIBUTED TBM: T=8, $s_1 \overrightarrow{\bigoplus} s_3$

Set #	s_1	s_2	$s_1 \bigotimes s_2$	$\bar{s}_{1,2}$	$s_1,\ s_2$
$\emptyset$	0.5362	0.341	0.7059382	0.23	0.58966
$\{a\}$	0.0102	0.011	0.000129	0.05	0.0021
$\{b\}$	0.453	0.647	0.2939322	0.71	0.40818
$\{a, b\}$	0.0006	0.001	0.0000006	0.01	0.00006

TABLE XXVI
DYNAMIC DISTRIBUTED TBM: T=9, $s_1 \overrightarrow{\bigoplus} s_2$

proposed for each scenario along with a theoretical characterization of its convergence properties.

Multi-agent systems represent an ideal abstraction of actual networks of mobile robots or sensor nodes that are envisioned to perform the most various kind of tasks. Therefore, we

Set #	s_1	s_2	$s_1 \bigotimes s_2$	$\bar{s}_{1,2}$	$s_1,\ s_2$
$\emptyset$	0.58966	0.341	0.73543462	0.341	0.58966
$\{a\}$	0.0021	0.011	0.00002586	0.011	0.0021
$\{b\}$	0.40818	0.647	0.26453946	0.647	0.40818
$\{a, b\}$	0.00006	0.001	0.00000006	0.001	0.00006

TABLE XXVII
DYNAMIC DISTRIBUTED TBM: T=10, $s_2 \overrightarrow{\bigoplus} s_4$

Set #	s_1	s_2	$s_1 \bigotimes s_2$	$\bar{s}_{1,2}$	$s_1,\ s_2$
$\emptyset$	0.58966	0.5362	0.814799044	0.5362	0.58966
$\{a\}$	0.0021	0.0102	0.000023292	0.0102	0.0021
$\{b\}$	0.40818	0.453	0.185177628	0.453	0.40818
$\{a, b\}$	0.00006	0.0006	0.000000036	0.0006	0.00006

TABLE XXVIII
DYNAMIC DISTRIBUTED TBM: T=11, $s_1 \overrightarrow{\bigoplus} s_3$

Set #	s_1	s_2	$s_1 \bigotimes s_2$	$\bar{s}_{1,2}$	$s_1,\ s_2$
$\emptyset$	0.58966	0.5362	0.814799044	0.5362	0.58966
$\{a\}$	0.0021	0.0102	0.000023292	0.0102	0.0021
$\{b\}$	0.40818	0.453	0.185177628	0.453	0.40818
$\{a, b\}$	0.00006	0.0006	0.000000036	0.0006	0.00006

TABLE XXIX
DYNAMIC DISTRIBUTED TBM: T=12, $s_3 \overrightarrow{\bigoplus} s_5$

believe that the proposed techniques make it possible to effectively apply the TBM in important engineering fields such as multi-robot systems or sensor networks, where the distributed collaborations among players is a critical and yet crucial aspect.

Future work will be mainly focused on the extension of the proposed technique on more complex topologies such as graphs with cycles. Indeed, whereas tree-like topologies can properly represent interaction among static sensors, the use of cyclic structures better describes the interaction among mobile units.

ACKNOWLEDGE

The authors would like to thank prof. Alessandro Saffiotti and prof. Giovanni Ulivi for their useful conversation.

REFERENCES

[1] M. R. Ghahroudi and R. Sabzevari, *Sensor and Data Fusion*, N. Milisavljevic, Ed. InTech, 2009.

[2] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer, 2001.

[3] Y. Xiang, "A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication," *Artif. Intell.*, vol. 87, pp. 295–342, November 1996. [Online]. Available: http://portal.acm.org/citation.cfm?id=240991.240999

[4] X. Yang and V. Lesser, "On the role of multiply sectioned bayesian networks to cooperative multiagent systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 33, no. 4, pp. 489 – 501, july 2003.

[5] J. Schwertfeger and O. Jenkins, "Multi-robot belief propagation for distributed robot allocation," in *6th IEEE International Conference on Development and Learning*, 2007.

[6] J. Schiff, E. Sudderth, and K. Goldberg, "Nonparametric belief propagation for distributed tracking of robot networks with noisy inter-distance measurements," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[7] F. V. Jensen and T. D. Nielsen, *Bayesian Networks and Decision Graphs*, 2nd ed. Springer Publishing Company, Incorporated, 2007.

[8] G. Shafer, *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976.

[9] P. Smets, "Belief functions versus probability functions," in *Uncertainty and Intelligent Systems*, ser. Lecture Notes in Computer Science, B. Bouchon, L. Saitta, and R. Yager, Eds. Springer Berlin / Heidelberg, 1988, vol. 313, pp. 17–24.

[10] ——, "Transferable belief model versus bayesian model," in *ECAI*, 1988, pp. 495–500.

[11] B. R. Cobb and P. P. Shenoy, "A comparison of bayesian and belief function reasoning," *Information Systems Frontiers*, vol. 5, pp. 345–358, December 2003.

[12] D. Ghosh, D. Pados, R. Acharya, and J. Llinas, "On dempster-shafer and bayesian detectors," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 36, no. 5, pp. 688 –692, sept. 2006.

[13] P. Smets, "Information content of an evidence," *Int. J. Man Machine Studies*, vol. 19, pp. 33–43, 1983.

[14] P. Smets and R. Kennes, "The transferable belief model," *Artificial Intelligence*, vol. 66, pp. 191–234, 1994.

[15] R. Kennes and P. Smets, "Fast algorithms for dempster-shafer theory," in *Uncertainty in Knowledge Bases*, ser. Lecture Notes in Computer Science, B. Bouchon-Meunier, R. Yager, and L. Zadeh, Eds. Springer Berlin / Heidelberg, 1991, vol. 521, pp. 14–23.

[16] Y. Denoeux, "Approximating the combination of belief functions using the fast moebius transform in a coarsened frame," *International Journal of Approximate Reasoning*, vol. 31, no. 1-2, pp. 77–101, 2002.

[17] P. Smets, "The application of the transferable belief model to diagnostic problems," *Int. J. Intelligent Systems*, vol. 13, pp. 127–157, 1998.

[18] F. Delmotte and P. Smets, "Target identification based on the transferable belief model interpretation of Dempster-Shafer model." *IEEE Transactions on Systems, Man and Cybernetics*, vol. A 34, pp. 457–471, 2004.

[19] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401 – 420, march 2006.

[20] Y. Peng, R. Freeman, and K. Lynch, "Multi-agent coordination by decentralized estimation and control," *Automatic Control, IEEE Transactions on*, vol. 53, no. 11, pp. 2480 –2496, dec. 2008.

[21] X. Liu, W. Lu, and T. Chen, "Consensus of multi-agent systems with unbounded time-varying delays," *Automatic Control, IEEE Transactions on*, vol. 55, no. 10, pp. 2396 –2401, oct. 2010.

[22] H.-T. Zhang, C. Zhai, and Z. Chen, "A general alignment repulsion algorithm for flocking of multi-agent systems," *Automatic Control, IEEE Transactions on*, vol. 56, no. 2, pp. 430 –435, feb. 2011.

[23] R. Beard, J. Lawton, and F. Hadaegh, "A coordination architecture for spacecraft formation control," *Control Systems Technology, IEEE Transactions on*, vol. 9, no. 6, pp. 777 –790, nov 2001.

[24] J. Liu and B. Hu, "On emergent complex behaviour, self-organised criticality and phase transitions in multi-agent systems: autonomy oriented computing (aoc) perspectives," *International Journal of Modelling, Identification and Control*, vol. 3, no. 1, pp. 3–16, 2008.

[25] Z. Elouedi, K. Mellouli, and P. Smets, "Assessing sensor reliability for multisensor data fusion within the transferable belief model," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 1, pp. 782 – 787, February 2004.

[26] H. Guo, W. Shi, and Y. Deng, "Evaluating sensor reliability in classification problems based on evidence theory," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 5, pp. 970 –981, October 2006.

[27] P. Smets, "Analyzing the combination of conflicting belief functions," *Inf. Fusion*, vol. 8, pp. 387–412, October 2007.

[28] X.-S. Si, C.-H. Hu, J.-B. Yang, and Q. Zhang, "On the dynamic evidential reasoning algorithm for fault prediction," *Expert Syst. Appl.*, vol. 38, pp. 5061–5080, May 2011.

[29] L. A. Zadeh, "On the validity of dempster rule of combination," in *In Memo M 79/24*. Univ. of California, Berkeley, 1979.

[30] P. Smets, "The combination of evidence in the transferable belief model," *IEEE TPAMI*, vol. 12, no. 5, pp. 447–458, 1990.

[31] D. Lucarelli and I.-J. Wang, "Decentralized synchronization protocols with nearest neighbor communication," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 62–68.

[32] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robot. Auton. Syst.*, vol. 56, pp. 324–333, April 2008.

[33] E. M. R. Aragues and C. Sagues, "Consistent data association in multi-robot systems with limited communications," in *Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[34] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 2508–2530, 2006.

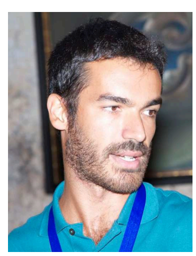
[35] K. Johansen, U. Jørgensen, S. Nielsen, S. Nielsen, and S. Skyum, "A distributed spanning tree algorithm," *Distributed Algorithms*, 1988.

[36] Y. Dalal, "A distributed algorithm for constructing minimal spanning trees," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 3, pp. 398–405, 1987.

[37] P. Smets, "The concept of distinct evidence," in *IPMU*, 1992, pp. 89–94.

Andrea Gasparri received the Graduate degree (cum laude) in computer science in 2004 and the Ph.D. degree in computer science and automation in 2008, both from the University of Rome "Roma Tre", Rome, Italy. He is currently an Assistant Professor at the Department of Computer Science and Automation, University of Rome "Roma Tre". His current research interests include mobile robotics, sensor networks, and, more generally, networked multi-agent systems.

Flavio Fiorini was born in Rome (Italy) in 1985. On May 26, 2010 he took the Laurea degree in Autyomation Engineering at University of Rome "Roma Tre". He is currently working at Logofive Srl (TO).



Maurizio Di Rocco received the Graduate degree (cum laude) in computer science in 2006 and the Ph.D. degree in computer science and automation in 2010, both from the University of Rome "Roma Tre", Rome, Italy. His research is focused on embedded systems, mobile robotics and distributed data fusion.

Stefano Panzieri was born in Roma (Italy) in 1963. He received the "Laurea" degree in Electronic Engineering in 1989 and the Ph.D. in Systems Engineering in 1994, both from the University of Roma "La Sapienza". Since February 1996 he is with the "Dipartimento di Informatica e Automazione" (D.I.A.) of University of Rome "Roma Tre", where currently is Associate Professor. His research interests are in the field of industrial control systems, mobile robots and critical infrastructures modeling.