Robust coordinated reinforcement learning for MAC design in sensor networks

Eleni Nisioti and Nikolaos Thomos *Senior Member, IEEE*

***Abstract*—In this paper, we propose a medium access control (MAC) design method for wireless sensor networks based on decentralized coordinated reinforcement learning. Our solution maps the MAC resource allocation problem first to a factor graph, and then, based on the dependencies between sensors, transforms it into a coordination graph, on which the max-sum algorithm is employed to find the optimal transmission actions for sensors. We have theoretically analyzed the system and determined the convergence guarantees for decentralized coordinated learning in sensor networks. As part of this analysis, we derive a novel sufficient condition for the convergence of max-sum on graphs with cycles and employ it to render the learning process robust. In addition, we reduce the complexity of applying max-sum to our optimization problem by expressing coordination as a multiple knapsack problem (MKP). The complexity of the proposed solution can be, thus, bounded by the capacities of the MKP. Our simulations reveal the benefits coming from adaptivity and sensors' coordination, both inherent in the proposed learning-based MAC.**

***Index Terms*—Medium Access Control, Q-learning, Coordination Graphs, Irregular Repetition Slotted ALOHA, wireless sensor networks, POMDP, Max-sum algorithm**

I. Introduction

Wireless sensor networks (WSNs) have drawn the attention of the research community due to their wide applicability in environmental monitoring and object tracking. The design of efficient WSNs is challenging because of limitations associated with their operation, such as their ad hoc deployment, dynamic and self-configuring topology, uncertain and changing channel conditions, large size and lack of a centralized point of control. Furthermore, sensors have limited computational power, battery capacity and transmission range, traits that impose bounded complexity in systems utilizing WSNs [1].

As shared wireless resources, such as the common communication channel, are restricted, it is necessary to orchestrate the access of sensors to them. The design of a MAC protocol aims at improving the use of the common channel by optimizing the transmission strategies of sensors. Due to the irregular nature of communication and the need for efficient utilization of the channel, contention-based MAC protocols are traditionally preferred to the more conservative family of Time Division Multiple Access protocols [1], [2]. However, when sensors attempt to transmit their packets simultaneously, a collision occurs, and their transmissions fail. This degrades the network's efficiency both in terms of achieved throughput and energy consumption, as sensors need to dedicate battery, memory and computational resources to re-transmitting lost packets. It is, thus, necessary to employ mechanisms for resolving collisions. Furthermore, configurability and low complexity in implementation are essential when designing MAC solutions for resource-constrained, dynamic WSNs. These traits are inherent in probabilistically defined MAC protocols, while MAC protocols that require a deterministic transmission plan impose prohibitive memory requirements. In this work, we optimize Irregular Repetition Slotted ALOHA (IRSA) [3], a state-of-the-art probabilistic protocol that employs successive interference cancellation (SIC) to resolve collisions. IRSA is ruled by the employed degree distribution, a probability distribution that describes how many replicas[1] of the packets available to each sensor should be transmitted in each frame. The work in [3] proves that IRSA can approach optimal throughput in asymptotic settings, by relating SIC to the process of iterative erasure decoding of graph-based codes. Nevertheless, IRSA performs poorly in dynamic environments. This is because the theoretical analysis and optimization framework in [3] do not take into account the network topology and channel conditions, and assume frames of infinite duration. The latter imposes unnecessary delays, as sensors need to wait for the next frame to initiate a transmission and, therefore, renders real-time communication impossible.

Reinforcement learning is often leveraged to equip MAC protocols with adaptivity [1], [4], as it is well-suited for deriving probabilistic policies based on the effect of sensors' actions, such as collisions caused by transmission. However, the application of reinforcement learning in WSNs faces various challenges. Partial observability of the network's state and the lack of a centralized point of control call for decentralized solutions. Also, non-stationarity, due to the time-varying channel conditions and network topology, as well as decentralization, suggests that learning algorithms should be model-free. The above characteristics motivated us to base our modeling of WSNs on the decentralized POMDP framework, which we appropriately modify to derive the framework of Groupwise Dependent Decentralized POMDPs (GDD-POMDPs). This framework views WSNs as groups of sensors and accounts for MAC characteristics that relate to inter-group independence and intra-group observability, which are essential for proving that our learning algorithm is optimal.

The complexity that the aforementioned learning-related challenges introduce can be reduced by exploiting the locality of interaction inherent in WSNs, imposed by the restricted

Manuscript received December 15, 2018; revised April 7, 2019; accepted May 20, 2019. This work was partly funded by the European Union Horizon 2020 research and innovation programme under Marie Sklodowska-Curie Research and Innovation Staff Exchange grant through the project RECENT (grant agreement No. 823903). E. Nisioti and N. Thomos are with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, United Kingdom (e-mail:e.nisioti, nthomos@essex.ac.uk).

[1]A replica is an identical copy of a packet.

transmission range of sensors. Interactions of sensors imply a need for coordination, particularly for MAC design where: (i) decisions require information non-local to the sensors, (ii) limited computational and battery resources prohibit centralized solutions, and, (iii) sensors have different characteristics and degrees of observability and, thus, different perspectives on the resource allocation task. Coordination graphs (CGs) [5] can be a useful tool for MAC design due to their ability to leverage the structure of an optimization problem by exploiting the observation that, often, only a few agents interact with each other. This allows the decomposition of a coordination problem into simpler problems without losing optimality. In a CG, each node represents an agent, and agents are connected to each other when there is a coordination dependency, e.g., when a collision between packets occurs in a MAC protocol. Each dependency is associated with a group payoff function, which assigns a specific value to every possible action combination of the involved agents. The global payoff function of the network equals the sum of all group payoff functions.

To deal with the drawbacks mentioned above of existing MAC solutions, we formulate MAC design as a multi-agent system, where agents learn how to transmit their packets using reinforcement learning. Specifically, we assume that each sensor is an agent aiming at maximizing the throughput of the sensor network. Although our framework is presented for MAC design, it is generic and can be employed, with some adaptation, for other resource allocation problems in WSNs. In order to avoid the sub-optimal performance that independent learners are associated with [6], as well as the complexity and impracticality of centralized solutions, we adopt a coordinated learning approach, where we leverage the particular structure of the coordination problem by employing CGs. In our framework, sensors learn by using a Q-learning based algorithm, where the global Q-function is decomposed into a summation of Q-functions, each one corresponding to a group in the CG. Our approach can be combined with a variety of reinforcement learning algorithms, but Q-learning proves to be a good choice, as it is a model-free and intuitive algorithm that has already been successfully combined with CGs [7]–[9]. In order to derive the optimal actions, max-sum, an algorithm for approximate distributed probabilistic inference, is applied to the CG. To the best of our knowledge, this is the first attempt to optimize resource management in sensor networks using coordinated learning. In particular, in this work, we model our MAC optimization problem under the IRSA protocol and design the degree distribution of IRSA. Actions in our POMDP formulation correspond to the number of replicas sent by each sensor. We follow the observation in [3] that, in IRSA, transmissions can be represented by a bipartite graph and employ this graph to derive the CG, which is updated at the beginning of each frame. This approach differs significantly from works where the connectivity of the WSN is not provided but designed to improve inference [10].

As WSNs operate under restricted resources, we need to reduce further and bound the complexity of coordination. To achieve this, we formulate the optimization objective of max-sum as a multiple knapsack problem (MKP) [11], where sensors are removed from groups to satisfy the complexity constraints. This is done by eliminating edges on the CG, while constraints are controlled by the capacities of the knapsacks. We should note that this formulation is generic and can be used to achieve the desired balance between the sparsity of a bipartite graph and the quality of approximate inference. By appropriately redefining weights, the MKP can account for different types of constraints, such as the battery lifetime of sensors.

Furthermore, we prove that our Q-learning based algorithm converges to the optimal solution under the GDD-POMDP framework. Thereafter, we derive a sufficient condition for the convergence of max-sum, based on the work in [12], which contains an analysis of the convergence of sum-product, an algorithm that, along with max-sum, belongs in the family of belief propagation algorithms. This condition permits us to a priori evaluate whether max-sum will converge on a specific CG. Due to this mechanism, coordinated learning is rendered robust, as sensors can choose to coordinate only when convergence is guaranteed, while independent learning can be performed otherwise. In this way, sensors can reduce their energy consumption without degrading the overall throughput. We believe that our analysis is an important step towards addressing the convergence uncertainty inherent in non-stationary learning environments.

To summarize, our main contributions consist in:

- a novel solution for adaptive resource allocation in WSNs that requires minimum state information, where sensors coordinate to maximize the network's performance by employing CGs. Our learning algorithm is based on Q-learning and max-sum is applied on the CG to find the optimal sensors' transmission actions;
- the introduction of the GDD-POMDP framework, which has convergence guarantees for coordinated decentralized learning in sensor networks;
- the derivation of a sufficient condition for the convergence of max-sum;
- a novel technique to ensure bounded computational complexity of employing max-sum for coordination based on an MKP formulation;
- the improvement of IRSA in terms of throughput for small frame sizes.

The rest of the paper is structured as follows: Section II positions our work with respect to the related literature. Section III presents the MAC design problem under IRSA, as well as our optimization objective. In Section IV, an overview of CGs and max-sum is provided. Section V describes the proposed solution. Section VI introduces a technique for reducing its complexity, and, in Section VII, we formulate the framework of GDD-POMDPs and derive convergence guarantees for it. Section VIII evaluates our solution and offers insights into the effect of coordination and complexity. Finally, Section IX consolidates our observations.

II. Related work

Due to the increasing need for efficient communication over shared channels, contention-based MAC protocols have

seen a continuous improvement in their performance, that culminates in IRSA. The original Slotted ALOHA [2] is inappropriate for energy-constrained WSNs, as throughput cannot exceed 0.37 [3]. Diversity Slotted ALOHA [13] improves upon it by allowing sensors to transmit a pre-defined number of replicas of their packets. Contention Resolution Slotted ALOHA [14] manages to increase the achievable throughput to 0.55 by employing SIC to retrieve collided packets. In IRSA, the number of replicas is decided by sampling from a probability distribution, which is optimized in [3] using differential evolution so that throughput is increased. IRSA shows that combining SIC with diversity in the behavior of sensors, in the form of the number of transmitted replicas, enables throughput to asymptotically approach 0.97, which is significantly higher than the throughput achieved by previous schemes. More recently, in [15], Multi-armed Bandits are employed to optimize a prioritized version of IRSA in a non-asymptotic setting. The main drawback of using Multi-armed Bandits is that they are stateless, and, therefore, cannot capture sensors' characteristics, such as the battery level and the state of their buffers. In our previous work presented in [4], IRSA is optimized in a decentralized POMDP framework. However, independence among agents was assumed in that work, which may lead agents to converge to local optima.

Although traditional MAC solutions for WSNs are static and designed a priori [1], [3], recent advances in reinforcement learning have enabled adaptive resource management. For example, deep reinforcement learning is employed in [16], [17] for dynamic spectrum access, where classical Q-learning is coupled with function approximation to address the complexity arising due to the exponentially increasing learning space of WSNs. However, deep reinforcement learning cannot currently offer computationally feasible solutions for WSNs, as convergence to optimal strategies is an open issue, while training, which is required if the network conditions change significantly, has to be performed offline and centrally due to its computational complexity. Game-theoretic tools, such as Nash Equilibria and Pareto optimality, are an alternative approach to analyzing the learning dynamics in multi-agent systems [16], [17]. Nevertheless, their application in WSNs is problematic due to the existence of a large number of sensors, the continuous and stochastic nature of decision variables, and the restricted observability of agents. In order to ensure convergence to efficient operating points, techniques such as *common training* [16] require additional assumptions about the behavior of agents.

Early approaches to solve problems using CGs entailed high complexity due to adopting exact inference techniques, such as variable elimination (VE) [8], the complexity of which scales exponentially in the induced width of the graph. For example, in [8], the Q-value function is first decomposed into a linear summation of local functions and VE is employed to find the optimal joint action using a cost graph. Similarly to some extent to our robust learning approach, where independent learning is performed if the sufficient condition indicates that max-sum will not converge, in [18], coordination is avoided based on the problem's context, and VE is employed otherwise. The complexity introduced by VE can be avoided by adopting approximate inference algorithms for computing the maximum a posteriori configuration [7], [19], as they are anytime algorithms with limited communication overhead, that scales linearly with the number of agents. Max-sum, an approximate inference message-passing algorithm, is used in [7] to choose the optimal actions in a sensor network that employs Coordinated Q-learning for object tracking.

There are two major concerns when employing max-sum for coordination: the lack of optimality guarantees and the involved complexity. To overcome the fact that max-sum is not guaranteed to converge on arbitrary graphs [20], in [21], the bounded max-sum algorithm, a variant of max-sum with bounded worst-case distance from the optimal solution, is proposed. This algorithm reduces the coordination problem to a tree-structure by eliminating some of the problem's constraints in order to guarantee convergence. Similar to our work, the algorithm in [21] further attempts to reduce the complexity of max-sum. To this end, two techniques are presented: one for pruning dominated actions and a branch-and-bound technique for reducing the search space. As this technique aims at guaranteeing bounded quality of the solution, it differs from our MKP formulation, which focuses on maximizing the quality of the solution while ensuring bounded complexity.

Note that the behavior of max-sum on graphs with cycles is, to date, a widely unexplored area. However, the "folklore" is that failure of max-sum to converge leads to bad solutions, whereas convergence has been experimentally validated to, almost consistently, find the optimal maximum a posteriori probability [22], [23]. In [12], sufficient conditions for sum-product to converge are derived and, in [24], it is proven that, if max-product converges, the solution is a neighborhood optimum, i.e, it is optimal in the sub-space defined by the variables involved in a group, and, thus, outperforms algorithms that converge to local optima [24].

III. Problem description

Let us consider a network of C sensors collecting measurements from their environment and transmitting them to the core network for further process. The main bottleneck of this operation is the occurrence of collisions, as transmissions are performed through a common channel. In our work, time is divided into frames comprised of D time slots. We also assume that sensors have a buffer of finite size B, where packets are stored. At the beginning of a frame, each sensor attempts transmission of its packets in randomly selected slots, while a source injects F_t number of packets into its buffer with a probability p_f. Without loss of generality, we assume that D is constant and each sensor transmits at most one packet per frame. The channel is characterized by its normalized traffic, defined as $G = C/D$, which represents the average number of attempted packet transmissions per time slot. The quality of the MAC protocol is quantified by the normalized throughput T, which is defined as the probability of successful packet transmission per slot. When the transmission of a packet fails, it stays in the buffer for future re-transmission. When the number of stored packets exceeds the buffer capacity B, an overflow occurs and packets are lost.

In IRSA, a sensor has the capability of transmitting a variable number of replicas of a packet in the available time slots. To choose this number, IRSA samples the degree distribution $\Lambda(x)$, a polynomial probability distribution describing the probability Λ_l that a sensor transmits l replicas of its message at a particular frame, which is formally expressed as:

$$\Lambda(x) \triangleq \sum_{l=1}^{d} \Lambda_l x^l \tag{1}$$

where d is the maximum number of replicas a sensor is allowed to send. If one of the replicas is transmitted in a collision-free slot, then the packet is successfully received. Furthermore, IRSA substantially improves throughput by employing SIC [3], a technique that resolves collisions under the rationale that, if two replicas collide, they might still be recovered by removing the interference of a replica that has previously been successfully received.

The objective of our optimization is to select the values Λ_l in (1) so that the overall throughput T is maximized. As there is a direct relationship between the achievable throughput and the degree distribution, we express T in terms of $\Lambda(x)$. This dependence becomes obvious if one considers the waterfall effect [3] in IRSA. According to this phenomenon, there exists a threshold value G^* for the channel load G, that depends on $\Lambda(x)$, above which transmission fails with a probability bounded away from 0. Formally, the optimization objective we aim to solve can be cast as:

$$\text{Find: } (\Lambda^*(x)) : \arg\max_{\Lambda(x)} T(\Lambda(x)), \text{ s.t. } \sum_{l=1}^{d} \Lambda_l = 1. \tag{2}$$

Although SIC often successfully resolves collisions, it can fail if there are too many of them. This happens because the iterative algorithm employed gets stuck in cycles [3]. In IRSA, this can be avoided if sensors indirectly avoid each other, by transmitting a number of replicas that will result in the smallest probability of collision, as slots are assigned to packets uniformly at random. Therefore, our work equips sensors with the ability to coordinate their transmission policies in order to avoid simultaneously sending too many (unresolved collisions) or too few (underutilized channel) replicas. For example, in Fig. 1a, sensors 1 and 3 choose to transmit 3 and 4 replicas respectively, which are both large numbers considering a frame of 5 slots, presented in Fig. 1b. In this frame, the transmission will fail for all sensors, even though sensor 2 sent only one replica. This could lead all sensors to send a small number of replicas in the next frame and, thus, successfully transmit their packets.

IV. Coordination graphs and the max-sum algorithm

The problem presented in Section III can be seen, on a higher-level, as a group of agents that attempt to maximize the overall throughput by coordinating their packet replica transmissions. Specifically, each agent (sensor) i chooses an individual action a_i from a set $\mathcal{A}_i$, and the resulting joint action $a = \langle a_1, \cdots, a_C \rangle$ generates a payoff $f(a)$ for the network. In our problem, action a_i corresponds to the number of packet replicas sensor i sends and the payoff $f(a)$ to the overall throughput T. The aim of the coordination problem is to find the optimal joint action a^* that maximizes $f(a)$. The optimized $\Lambda(x)$ of (2) can then be estimated from the history of actions. An obvious approach to determine the optimal action is to consider all possible joint actions and select the one that maximizes $f(a)$. However, this approach quickly becomes impractical, since the joint action space grows exponentially with the number of sensors.

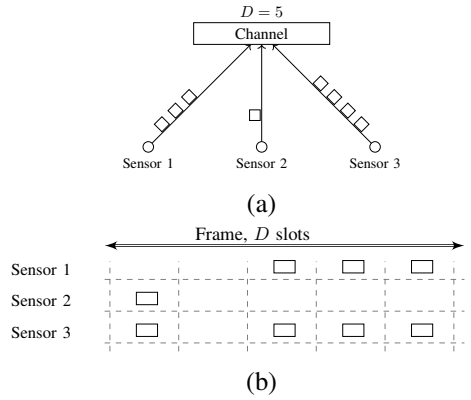


Fig. 1: Transmission under IRSA: (a) a sensor network consisting of three sensors that wirelessly transmit replicas of their packets to a common channel, and (b) transmissions of replicas in a frame.

Fortunately, coordination problems in WSNs exhibit the property that the payoff matrix $f(a)$ is sparse. This suggests that each agent is affected only by the decisions made by a small subset of the agents, as only sensors that collide have to coordinate their actions. In this paper, we consider the use of a CG to account for such dependencies. This allows us to decompose the global payoff function $f(a)$ into a linear combination of group payoff functions, each involving only a smaller number of agents. For example, a payoff function involving the three sensors of Fig. 1 can be decomposed as follows:

$$f(a) = f_{13}(a_1, a_3) + f_{23}(a_2, a_3) \tag{3}$$

We can map function $f(a)$ to a CG $(\mathcal{N}, \mathcal{L})$, as the one depicted in Fig. 2a. Each node in $\mathcal{N}$ represents an agent, while an edge in $\mathcal{L}$ indicates a coordination dependency (a collision that occurred in the previous time frame). Only connected agents have to coordinate their actions at any particular time instance. The global optimization problem is, thus, recast as a number of local coordination problems, each involving a subset of the total number of agents, that can be solved distributively. Thus, agents can find their optimal values independently of agents that do not participate in their local coordination problem. Sparsity of the CG is directly related to the complexity of coordination, as the computational complexity of max-sum scales exponentially with the number of variables on which the group payoff functions depend. Edges in a CG represent dependencies and, thus, increase the arity of group payoff functions. Note, however, that the number of exchanged messages varies linearly with the number of

agents. As such, the increase in complexity is not due to communication overhead, but should be attributed to the exponential growth of the search space [21].

Variable elimination is an approach to finding the optimal joint action a^* [5]. This algorithm eliminates the agents from the graph one by one, and always finds the optimal joint action. However, its execution time is non-deterministic, due to its dependence on the order of elimination [19], as well as impractical for large WSNs, as it increases exponentially with the induced width of the CG [19]. To avoid the complexity required to determine the optimal solution, in this work, we adopt the use of the max-sum algorithm, which performs approximate inference on the CG.

The coordination problem in Fig. 2a can be alternatively represented using a bipartite graph, like the one presented in Fig. 2b, where interactions (collisions) between agents (sensors) are now explicitly drawn. In the bipartite representation, nodes in the lower row are termed as variable nodes (VNs), and correspond to sensors, while the upper row, consisting of the check nodes (CNs), represents the shared network resources, which in our case are time slots. Bipartite graphs offer much richer problem representations than simple coordination graphs, as they can represent k-ary ($k \geq 1$) relationships. Hereafter, we denote VNs with lower case letters, CNs with uppercase, the set of CNs as $\mathcal{F}$ and the set of VNs as $\mathcal{V}$. Also, the neighborhood of a variable node i is denoted as $\mathcal{N}_i = \{I \in \mathcal{F},\ i \in \mathcal{N}_I\}$ and the neighborhood of a check node I as $\mathcal{N}_I = \{i \in \mathcal{V},\ I \in \mathcal{N}_i\}$. If we express the dependencies in terms of utilities, i.e. define a quantity $u_I(a_i)$, where $I \in \mathcal{F}$, that expresses the utility of agent i when interacting with other agents, then we get the equivalent, interaction-based bipartite graph, shown in Fig. 2c.

The max-sum algorithm can be applied on the bipartite graph of Fig. 2c to solve the inference problem of finding the value assignment of a set of variables that maximizes a factored probability distribution. Consider $|\mathcal{V}|$ discrete random variables x_i for $i \in \mathcal{V} := \{1, 2, \cdots, |\mathcal{V}|\}$, with x_i taking values in $\mathcal{X}_i$ and $|\mathcal{X}_i|$ the size of the space of variable's i values. Note that, in our problem formulation, the random variable x_i corresponds to the possible number of actions, i.e. $x_i \triangleq a_i$ and $\mathcal{X}_i = \{1, \cdots, d\}$. We are interested in calculating:

$$a^* = \arg\max_a p(a) = \arg\max_a \prod_{I \in \mathcal{F}} u_{\mathcal{N}_I}(a_i) \tag{4}$$

$$\equiv \arg\max_a \sum_{I \in \mathcal{F}} \ln u_{\mathcal{N}_I}(a_i) \tag{5}$$

where $u_{\mathcal{N}_I}$ is the utility function of CN I. Due to instabilities arising from the multiplication of potentially small quantities, the problem is formulated as a summation of logarithms. During the application of the max-sum algorithm, variable and check nodes exchange messages of the following form:

$$\begin{aligned} &\text{From VN } j \text{ to CN } I: \quad \tilde{\mu}_{j\to I}(a_j) = \sum_{J \in \mathcal{N}_j \setminus I} \mu_{J\to j}(a_j) \\ &\text{From CN } I \text{ to VN } i: \quad \tilde{\mu}_{I\to i}(a_i) = \max_{a_I \setminus i} \big[\ln(u_{\mathcal{N}_I}(a_I)) \\ &\qquad + \sum_{j \in \mathcal{I} \setminus i} \mu_{j\to \mathcal{I}}(a_j)\big] \end{aligned} \tag{6}$$

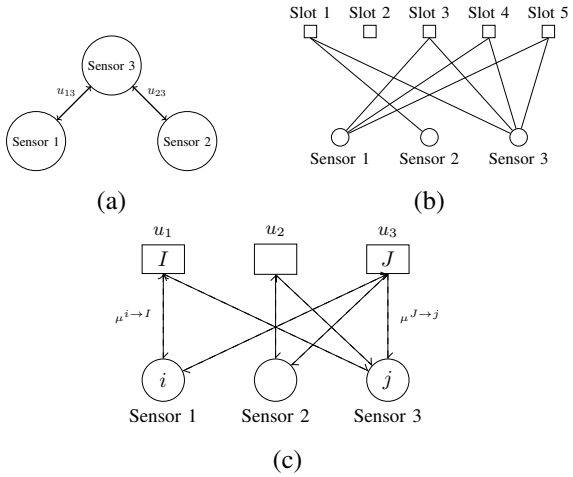


Fig. 2: Representing coordination in the sensor network: (a) the simple coordination graph, (b) the interaction-based bipartite graph (each check-node represents a slot), and (c) the utility-based bipartite graph (each check node computes the utility (u_I) of a variable node v_i that belongs to group I).

where $\mu(\cdot)$ corresponds to the current message and $\tilde{\mu}(\cdot)$ to the updated message, which will be used in the next iteration.

It is known that, for tree-structured graphs, max-sum converges to the optimal solution within a finite number of iterations [7]. Although it also empirically exhibits good performance for graphs with cycles [7], [23], there are no guarantees for convergence in this case. In Section VII, we derive a sufficient condition for max-sum to converge for arbitrary graphs and employ it to devise an optimal learning algorithm, as failure to converge is generally associated with solutions of bad quality [12], [22].

V. Coordinated reinforcement learning based MAC

In this section, we map the problem, as presented in Section III, and our underlying assumptions into the proposed learning framework. In our formulation, each sensor is an agent that interacts with its environment (channel and sensor network) by performing actions (number of replicas to send), accepts rewards (negation of number of packets in the transmission buffer) and makes observations (number of packets in its buffer). Partial observability in our framework arises due to the inability of sensors to observe the underlying global state of the network, denoted as s. Instead of employing the framework of Belief MDPs [25], in our setting, we use an approximation to beliefs based on a fixed history window of size w. Therefore, agents' state consists of a finite set of successive observations. The mathematical formulation of this finite-history POMDP for a sensor is:

$$\begin{aligned} \text{Observation:}\quad & \omega^t = b^t \qquad \text{Reward:}\quad r^t = -b^t \\ \text{Action:}\quad & a^t = l^t \qquad \text{State:}\quad \vec{h}^t = \langle \omega^{t-w+1}, \cdots, \omega^t \rangle \end{aligned}$$

where b^t is the number of packets in the buffer at time t and l^t is the number of replicas to send. For the sake of simplicity, we have omitted the sensor index from the above variables.

This definition of rewards and observations only requires information that is local to the sensors, in particular, the number of packets in their buffers. Note that rewards can implicitly provide information about the success of transmission, as packets are added to the sensors' buffer due to a packet arrival or stay in the buffer because of a transmission failure. Simultaneously, rewards do not depend only on the current success of transmission, but motivate sensors to avoid an overflow of their buffers, as this would result in packet loss. Similarly, for the observations, a sensor can discriminate states based on how full its buffer is, which leads the sensor to adapt its strategy to low and high data traffic.

Our goal is to compute a joint policy $\pi^*(\vec{h}, a)$ that maximizes the total expected reward of all agents over a finite horizon τ, termed as the optimal policy. Q-learning [26] is an approach to determining $\pi^*(\vec{h}, a)$, which employs the following update mechanism to estimate the value of a history-action pair:

$$Q(\vec{h}^t, a^t) = (1-\alpha)Q(\vec{h}^t, a^t) + \alpha[r^t + \gamma \max_a Q(\vec{h}^{t+1}, a)] \quad (7)$$

where α is the learning rate, dictating how quickly new acquired information overrides past one, and γ is the discount factor, determining how much future information is discounted. We henceforth refer to $Q(\vec{h}^t, a^t)$ as the Q-function or Q-table.

Depending on the level of communication we allow between sensors, Q-learning can: (i) prohibit communication and leave sensors to learn independently, and (ii) assume perfect communication among sensors and, thus, solve the problem jointly. Coordinated learning achieves a compromise between these two extreme approaches by generally assuming independence between agents and requiring communication only when a collision occurs, i.e., based on the CG. Note that communication in our setting happens off-line and messages exchanged among sensors regard only their actions. Next, we present an algorithm, based on Q-learning for POMDPs, employed by each sensor to learn by coordinating its actions using the max-sum algorithm. The process of finding the optimal actions consists of the following steps:

1) At the beginning of each frame a bipartite graph, of the form presented in Fig. 2b, is built based on the collisions that occurred in the previous time frame.
2) The graph is converted to a utility-based representation (Fig. 2c). In this graph, each VN is a sensor and each CN involves its Q-function. This suggests that, in our setting, the utility functions, as described in Section IV, are mapped to the Q-functions, i.e. $u_{\mathcal{N}_I}(a_i) \triangleq Q_l(\vec{h}_l, a_l)$, where $l \in \mathcal{L}$ indicates the index of a group of sensors, $\mathcal{L}$ refers to the set of groups, and I is the CN all sensors in group l are connected to.
3) Max-sum is applied on the graph to distributively calculate the optimal joint action of the global Q-function, defined as:

$$a^* = \arg\max_a Q(\vec{h}, a) \quad (8)$$

Note that, in the preceding equation, we dropped time index t, as max-sum is performed independently for each frame. Thus, the term $\vec{h}$ remains constant throughout the application of max-sum. The groupwise decomposition of the global Q-function is expressed as:

$$\hat{Q}(\vec{h}, a) = \sum_{l \in \mathcal{L}} Q_l(\vec{h}_l, a_l) \quad (9)$$

Based on (9), the update rule presented in (7) can be expressed using $\hat{Q}(\vec{h}, a)$ as:

$$\sum_{l \in \mathcal{L}} Q_l(\vec{h}_l^t, a_l^t) = (1-\alpha) \sum_{l \in \mathcal{L}} Q_l(\vec{h}_l^t, a_l^t) + \quad (10)$$

$$\alpha[\sum_{l \in \mathcal{L}} r_l^t + \gamma \max_a \hat{Q}(\vec{h}^{t+1}, a)] \quad (11)$$

Note that term $\max_a \hat{Q}(\vec{h}^{t+1}, a)$ cannot be further decomposed into a sum of local discounted future rewards, as this would require knowledge of the joint optimal action. Therefore, we define:

$$a^* = \arg\max_a \hat{Q}(\vec{h}^{t+1}, a) \quad (12)$$

$$\max_a \hat{Q}(\vec{h}^{t+1}, a) = \hat{Q}(\vec{h}^{t+1}, a^*) = \sum_{l \in \mathcal{L}} Q_l(\vec{h}_l^{t+1}, a_l^*) \quad (13)$$

where the last equality is due to (9).

The update mechanism for each group is thus:

$$Q_l(\vec{h}_l^t, a_l^t) = (1-\alpha)Q_l(\vec{h}_l^t, a_l^t) + \alpha[r_l^t + Q_l(\vec{h}_l^{t+1}, a_l^*)] \quad (14)$$

where we employ max-sum to determine a_l^*. We, then, perform the optimal actions a^* (or an exploratory action) and, based on the received rewards, update the group Q-functions Q_l.

VI. Complexity reduction

A. Motivation

Despite exploiting locality of interaction, the application of max-sum can still be prohibitive for energy-constrained WSNs, if the CGs are not sparse enough. In particular, as the channel load or frame size increases, collisions also increase in number, especially at the beginning of the learning process, when agents have not yet learned how to avoid transmitting in a way that will lead to unresolved collisions. We are, thus, still in need of techniques that will reduce the search space of max-sum without affecting the quality of its solution.

We start with the observation that, often, a small fraction of the original variables involved in a complex mathematical problem is required to determine the optimal solution [11]. In this paper, we use column generation [27], which exploits this observation by expressing the problem as an integer program and considering only a subset of its original variables. A common approach is to formulate the optimization objective as a multiple knapsack problem [11]. The intuition behind using column generation is that an agent can, under circumstances, ignore some of the agents it collided with, and still compute its optimal action correctly. We can, thus, reduce the complexity of the original problem by pruning some of the variables involved in Q_l.

Definition 1. The 0-1 MKP is: given a set of N items and M knapsacks ($M \leq N$), where each item has a value p_j and a weight w_j, and each knapsack has a capacity c_i, to select M

disjoint subsets of items, that can be assigned to a knapsack whose capacity is no less than the total weights of items in it, so that the total profit of the selected items is maximized. Formally:

$$\max \sum_{i=1}^{M}\sum_{j=1}^{N} p_j x_{ij} \tag{15}$$

$$\text{subject to} \quad \sum_{j=1}^{N} w_j x_{ij} \leq c_i, \quad i \in \mathcal{M} = \{1, \cdots, M\} \tag{16}$$

$$\sum_{i=1}^{M} x_{ij} \leq 1, \quad j \in \mathcal{N} = \{1, \cdots, N\} \tag{17}$$

$$x_{ij} \in \{0, 1\}, \quad i \in \mathcal{M},\ j \in \mathcal{N} \tag{18}$$

$$\text{where} \quad x_{ij} = \begin{cases} 1, \text{ if item } j \text{ is assigned to knapsack } i \\ 0 \text{ otherwise} \end{cases} \tag{19}$$

B. Multiple knapsack formulation of max-sum

In our setting, each Q_l corresponds to a knapsack and each agent to an item. Our objective is to determine a_l for each group $l \in \mathcal{L}$, so that the sum of the knapsacks, which denotes the overall utility of the sensor network, is maximized. The MKP will determine which agents to include in each Q-table. Note that condition (17) must be modified in our case, as agents can be included simultaneously in different knapsacks. We, thus, replace (17) with $\sum_{i=1}^{M} x_{ij} \leq M,\ j \in \mathcal{N}$, which forces an agent to be in a maximum of M knapsacks.

We define the value of an agent j in (15) as:

$$p_j = \max_{a_j} Q_j(\vec{h}_j, a_j), \quad a_j \in \{1, \cdots, d\} \tag{20}$$

where d is the maximum allowed number of replicas. This definition suggests that an agent is evaluated based on the maximum value of its local Q-table, which can be interpreted as the maximum contribution this agent expects to have to the maximization of the group Q-table. Note that this is not equal to $Q_j(\vec{h}_j, a_l^*[j])$, i.e., the component of the globally optimal solution that corresponds to agent's j action. This is because $a_l^*[j]$ is not the same with the action of j that maximizes its local Q-table, as the effect that this action will have on the different Q_l agent j participates in, is not considered. Thus, the solution provided by solving the MKP will not be globally optimal and max-sum should still be applied.

In order to define the weights w_j, we should measure how the participation of an agent in the Q-function of a group increases the computational complexity of applying max-sum. In particular, in this paper, we measure complexity as the time required until the convergence of max-sum. To determine this time we make use of the fact that max-sum exhibits a linear convergence rate, as we prove in Section VII. The convergence rate is governed by $|x^* - x^0|$, where x^* is the optimal value assignment and x^0 is the value max-sum is initialized with. We, therefore, know that, the further from the optimal solution max-sum starts, the more time it will need to converge. We can, thus, define the weight as $|x^* - x^0|$. Note that variables x^* and x^0 correspond to probability distributions over the agents' decision variables, as they are the messages sent from check to variable nodes during the application of max-sum. Formally:

$$\begin{aligned} x^0 &= \mu_i^n(a_i) \\ &= \sum_{I \in N_i} \mu_{I \to i}(a_i), \quad a_i \in \{1, \cdots, d\}, \quad n = 0 \end{aligned} \tag{21}$$

$$\begin{aligned} x^* &= \mu_i^n(a_i) \\ &= \sum_{I \in N_i}^{n} \mu_{I \to i}(a_i), \quad a_i \in \{1, \cdots, d\},\ n = N_{\max} \end{aligned} \tag{22}$$

where n is the index of the max-sum iteration, and $N_{\max}$ is the maximum allowed number of iterations. With a slight abuse of notation, we refer to the messages received by VN i as $\mu_i(a_i)$. Note that (22) is valid only when max-sum converges to the optimal solution. Although restricting the number of iterations in graphs with cycles can lead to sub-optimal solutions, if $N_{\max}$ is chosen to be appropriately high, it will not significantly affect the quality of the solution. This is due to the empirical observation that, if max-sum converges, this happens within the first few iterations [23].

We calculate the distance between x^* and x^0 as their Kullback - Leibler (KL) divergence, i.e:

$$w_j = D_{KL}(x^* || x^0) = \sum_{i=1}^{d} x^*(i) \log \frac{x^0(i)}{x^*(i)} \tag{23}$$

Our choice of the KL divergence was motivated by the observation in [28] that this measure naturally describes the lack of fit of an approximation of a distribution when preferences (beliefs) are expressed by a logarithmic function. The employed max-sum algorithm is an example of such a case, as beliefs are the logarithms of utilities. Furthermore, the additive property of the KL divergence is useful, both for representing the collective belief of agents, as well as for summing the weights of the items in a knapsack to determine the total weight assigned to it.

The value of x^* in (22) could be found by employing the max-sum algorithm, but the calculation of weights should not require this. We, therefore, approximate x^* based on the system's Gibbs free energy [29], an alternative approach to finding the solution of (4). Instead of employing message-passing, this solution computes the optimal probability distribution x^* by minimizing the KL divergence between x^* and the joint probability distribution, given by:

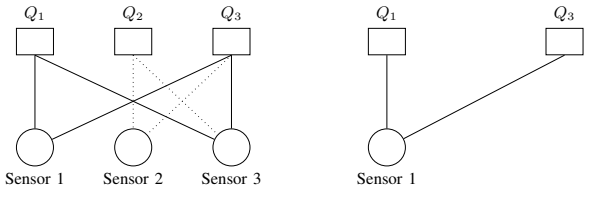
$$p(a) = \frac{1}{Z} \exp \sum_{I \in \mathcal{F}} \ln u_{\mathcal{N}_I}(a_{\mathcal{N}_I}) \tag{24}$$

where Z is a normalizing constant to ensure that $p(\cdot)$ sums to 1. Thus, x^* can be found by solving the following optimization problem:

$$\min \sum_{a_i} q(a_i) \log p(a_i) - \sum_{a_i} q(a_i) \log q(a_i) \tag{25}$$

$$\text{where} \quad q(a_i) = \sum_{J \in N_i} \mu_{J \to i}(a_i) \text{ and } p(a_i) \tag{26}$$

$$= -\frac{1}{Z} \exp \sum_{c \in C_i} \ln u_{\mathcal{N}_I}(a_{\mathcal{N}_I}) \tag{27}$$



(a) (b)

Fig. 3: (a) Agents 1 and 3 have collided, so they participate in each other's Q-function. The MKP needs to decide whether sensor 1 will be included in Q_3. (b) We form the sub-graph for calculating the weight of sensor 1 by assuming that Q_1 is independent from the messages from sensor 3.

Fig. 3 illustrates how the weight of an agent is determined. In this example, when calculating agent's 1 effect on Q_3, we ignore agent's 3 effect on Q_1, otherwise self-referentiality would not allow us to solve the problem. As the weight of an agent j depends on the knapsack j it is evaluated for, we henceforth include both indexes when denoting a weight (w_{ij}), and define w_j as the average weight of agent j.

Finally, we define the capacity of a knapsack, i.e. the time available to a Q-table to converge, based on the problem's feasibility constraints. We know that, in general, $c_i \geq \min w_{ij},\ \forall j \in \mathcal{N}$ and $w_{ij} \leq \max c_i,\ \forall i \in \mathcal{M}$. If a knapsack violates the first constraint, it can be ignored because no items can be added to it. Similarly, if an agent violates the second constraint, then it does not fit anywhere, and can, therefore, be eliminated. By sampling c_i from the range $[w_{ji}, \sum_{j\in\mathcal{M}} w_{ij})$, we ensure that a Q-table can at least include the agent it is associated with and that, not all the agents can fit in it.

C. Solving the MKP

We solve (15) by obtaining a tight bound on the optimal solution using the Lagrangian relaxation method [30]. We do not employ Branch-and-bound algorithms [11], although they offer an exact solution to the MKP problem. This is because their high computational complexity would defeat the original purpose of reducing the complexity of coordination. Besides, even if these algorithms were used, the final solution would still have to be found by the max-sum algorithm, as explained in Section VI-B.

The Lagrangian relaxation of (15) can be formulated as:

$$L(MKP,\lambda) = \max \sum_{i=1}^{M}\sum_{j=1}^{N} \tilde{p}_j x_{ij} + M\sum_{j=1}^{N}\lambda_j \tag{28}$$

$$\text{where} \quad \tilde{p}_j = p_j - \lambda_j, \quad j \in \mathcal{N},\ i \in \mathcal{M} \tag{29}$$

Similarly to [11], we find the optimal dual variables λ_j associated with the constraints in (17):

$$\lambda_j = p_j - w_j \frac{p_c}{w_c} \quad \text{if } j < c, \text{ and 0 otherwise} \tag{30}$$

where c denotes the critical item. This corresponds to the first item that does not fit in the knapsack, if we consecutively insert items in decreasing order of value per weight unit, and is formally defined as:

$$c = \min\left\{j : \sum_{i=1}^{j} w_{ij}|N_i| > \sum_{k=1}^{M} c_k\right\} \tag{31}$$

where $|N_i|$ denotes the number of neighbors of VN i. Note that our definition of the critical item differs from the classical one [11] due to the way that capacities are defined in our setting.

The relaxed problem can be subsequently decomposed into a series of independent single knapsack problems of the form:

$$\max \sum_{i=1}^{M}\sum_{j=1}^{N} \tilde{p}_j x_{ij} \tag{32}$$

$$\text{subject to} \quad \sum_{j=1}^{N} w_{ij}x_{ij} \leq c_i \quad \text{and} \quad x_{ij} \in \{0,1\},\ j \in \mathcal{N} \tag{33}$$

The optimal solution of (32) can be calculated as:

$$z(L(MKP,\lambda)) = \sum_{j\in J(\lambda)} \tilde{p}_j + M\lambda, \tag{34}$$

$$\text{where } J(\lambda) = \{j : p_j/w_j > \lambda\} \text{ and } \lambda = \sum_{j=1}^{N}\lambda_j. \tag{35}$$

VII. Optimality analysis

This section begins by introducing GDD-POMDPs, which map the properties of learning performed by WSNs into a mathematical framework for decision making. Subsequently, we derive guarantees for the convergence of coordinated Q-learning in this framework. The proof consists of two parts. First, we prove that the joint Q-function can be decomposed into a sum of group Q-value functions. Since max-sum is employed to choose the actions, we also have to ensure that its solution is optimal. Therefore, in the second part of the analysis, we derive a sufficient condition for the convergence of max-sum.

A. The GDD-POMDP framework

Based on our solution, as presented in Section V, sensors depend on each other only when their packets collide upon transmission. Dependence among agents regards both actions and states: two collided agents will have to coordinate their actions due to sharing the same utility function, and, they will also affect the state transition of each other. In our setting, transition and observation independence is not guaranteed for each agent (sensor). According to our formulation, it only concerns agents belonging to different groups, i.e., sensors whose packets have not collided. We refer to this framework as GDD-POMDPs and ascribe to it the property of groupwise observability [7].

Definition 2. A GDD-POMDP is defined as a tuple $\langle \mathcal{M}, \mathcal{S}, \mathcal{A}, T, R, \Omega, O, w, \xi^0\rangle$, where

$\mathcal{M} = \{1, \cdots, |\mathcal{M}|\}$ is the set of agent indices.

$\mathcal{S} = \times_{i\in\mathcal{M}}\mathcal{S}_i \times \mathcal{S}_u$. $\mathcal{S}_i$ refers to the local state of agent i. $\mathcal{S}_u$ refers to a set of uncontrollable states that are independent of the actions of the agents.

$\mathcal{A} = \times_{i\in\mathcal{M}}\mathcal{A}_i$, where $\mathcal{A}_i$ is the set of actions for each agent.
$\Omega = \times_{i\in\mathcal{M}}\Omega_i$ is the joint observation set.
$T(s'|s,a) = T_u(s'_u|s_u)\cdot\prod_{l\in\mathcal{L}} T_l(s'_l|s_l,s_u,a_l)$, is the transition probability function, where $a = \langle a_i,\cdots,a_M\rangle$ is the joint action performed in joint state $s = \langle s_i,\cdots,s_M\rangle$ and s_u is the current value of the uncontrollable state, the transitions of which are not affected by the actions of sensors, but are controlled by external factors (e.g. arrival/departure of sensors to/from a network). The transition probability distribution of the network is decomposable among groups of agents, indexed by l, with $l \in \mathcal{L}$, where $\mathcal{L}$ denotes the set of all groups. Note that we employ model-free learning and, thus, do not require a particular form for T_l. If $k = |l|$ agents with indices $\{i_1,\cdots,i_k\}$ are involved in a particular group l, then s_l denotes the state of group l, i.e. $s_l = \langle s_{l1},\cdots,s_{lk}\rangle$ and, similarly, $a_l = \langle a_{l1},\cdots,a_{lk}\rangle$. This decomposition models the transition independence between agents belonging to different groups.
$R = \sum_{i\in\mathcal{M}} R_i(s_i,s_u,a_i)$ is the immediate reward function. Thus, rewards are inherently local in this framework.
$O(\omega|s,a) = \prod_{l\in\mathcal{L}} O_l(\omega_l|s_l,s_u,a_l)$ is the observation probability function. This decomposition models the observation independence among groups.
w is the history window.
ξ^0 is the initial state distribution at time $t = 0$.

Definition 3. GDD-POMDPs are said to have groupwise observability if, $\forall l \in \mathcal{L}$, the set of observations $\omega_l = \langle\omega_{l1},\cdots,\omega_{lk}\rangle$, made by agents belonging in group l, fully determine the current uncontrolled state s_u, i.e., if $\forall l,\ \forall\omega_l,\ \exists s_u$: $Pr(s_u|\omega_l) = 1$.

In our framework, this property implies that, given the joint observation of a group $l \in \mathcal{L}$, ω_l, the observation and the transition probability function of the group l do not depend on actions and observations of agents in other groups.

B. Q-function decomposition

We base our analysis on [7], where the Q-function was also proven to be decomposable into a sum of group Q-functions. One notable difference between our setting and the one in [7] is that our framework is not ND-POMDPs, as in our case independence holds only for agents that do not belong in the same group. Another difference is that, in [7], the reward has the same decomposition as the Q-function and agents get their rewards by evenly distributing the group reward, whereas in our case rewards are individual to each sensor.

Theorem 1. For GDD-POMDPs with groupwise observability, under basic assumption of Q-learning and by means of update rule (14), $Q_l(\vec{h}_l,a_l)$ converges to the optimal $Q^*_l(\vec{h}_l,a_l)$ for all $l \in \mathcal{L}$, and thus, policy $\pi^*(\vec{h}) = \arg\max_a \sum_{l\in\mathcal{L}} Q^*_l(\vec{h}_l,a_l)$ is globally optimal.

In order to prove the above theorem, we first establish that a Q-function defined over states is decomposable. Then, we prove that a Q-function based only on histories of observations is also decomposable.

The Bellman equation for the global Q-function is:

$$Q(s^t,a^t) = R(s^t,a^t) + \gamma \sum_{s^{t+1},\omega^{t+1}} T^t_u T^t Q^{t*} \tag{36}$$

Equivalently, for the group Q-functions:

$$Q_l(s^t_l,a^t_l) = R(s^t_l,s^t_u,a^t_l) + \gamma \sum_{s^{t+1}_l,\omega^{t+1}_l} T^t_u T^t_l Q^{t*}_l, \quad \forall l \in \mathcal{L} \tag{37}$$

Recall that T^t_l cannot be further decomposed into a product of individual probability functions due to the absence of independence within a group.

In the case of Belief MDPs, we know that for the global Q-function:

$$Q(b^t,a^t) = \sum_{s\in S} b^t(s) Q(s^t,a^t) \tag{38}$$

If we replace continuous beliefs with finite histories of observations, then the above equations take the following form:

$$Q(\vec{h}^t,a^t) = \sum_{s\in S} b^t(s) Q(s^t,\vec{h}^t,a^t) \tag{39}$$

$$= \sum_{s_l\in\mathcal{S}_l, s_u\in\mathcal{S}_u} b^t(s_u,s_l) Q(s^t_l,\vec{h}^t_l,a^t_l) \tag{40}$$

$$\tag{41}$$

We can thus treat histories as a substitute for states in the Q-learning framework.

Lemma 2. In GDD-POMDPs, the global Q-function $Q(s^t,a^t)$ for any finite horizon τ is decomposable, that is:

$$Q(s^t,a^t) = \sum_{l\in\mathcal{L}} Q_l(s^t_l,a^t_l) \tag{42}$$

Proof. We prove the lemma by mathematical induction. For $t = \tau - 1$ we have by definition $Q(s^t,a^t) = R(s^t,a^t) = \sum_{l\in\mathcal{L}} r_l(s^t_l,a^t_l)$ and there is no future reward, as τ corresponds to the last iteration. Assume that for $1 \le t \le \tau - 1$ the global Q-function is decomposable, i.e. $Q(s^t,a^t) = \sum_{l\in\mathcal{L}} Q_l(s^t_l,\ a^t_l)$. Then, we have

$$\begin{aligned} Q(s^{t-1},a^{t-1}) &= R(s^{t-1},a^{t-1}) + \gamma \sum_{s^t,\ \omega^t} T^{t-1}_u T^{t-1} Q^* \\ &= \sum_{l\in\mathcal{L}} r^{t-1}_l + \gamma \sum_{s^t,\ \omega^t} T^{t-1}_u T^{t-1} \sum_{l\in\mathcal{L}} Q^*_l \\ &= \sum_{l\in\mathcal{L}} \Big[r^{t-1}_l + \gamma \sum_{s^t,\omega^t} p^{t-1}_u P^{t-1}_l Q^*_l \Big] \\ &= \sum_{l\in\mathcal{L}} Q^{t-1}_l \end{aligned}$$

where the last equality is valid by the assumption of mathematical induction. □

Lemma 3. In GDD-POMDPs with groupwise observability, the global Q-function $Q(\vec{h}^t,a^t)$ for any finite horizon τ is decomposable, that is:

$$Q(\vec{h}^t,a^t) = \sum_{l\in\mathcal{L}} Q_l(\vec{h}^t_l,a^t_l) \tag{43}$$

Proof.

$$Q(\vec{h}^t, a^t) = \sum_{s_u, s} b_u^t(s_u^t) b_s^t(s^t) \sum_{l \in \mathcal{L}} Q_l(s_l^t, a_l^t)$$
$$= \sum_{l \in \mathcal{L}} \Big[\sum_{s_u, s_l} b_l^t(s_u^t, s_l^t) Q_l(s_l^t, s_u^t, a_l^t) \Big]$$
$$= \sum_{l \in \mathcal{L}} Q_l(\vec{h}_l^t, a_l^t)$$

where the last equality arises from (42). Note that the decomposition of beliefs is valid due to Lemma 3.

Theorem 1 is a direct result of Lemmas 2 and 3. □

C. Convergence analysis of the max-sum algorithm

In this section, we derive a sufficient condition for the convergence of max-sum, based on the analysis in [12], where sufficient conditions for the sum-product algorithm were formulated. The intuition behind the proof is that the update mechanism for the messages can be expressed as a mapping in the vector space of messages. Thereafter, the conditions under which this mapping is a contraction can be derived, so that convergence to a fixed point is guaranteed. Our analysis can be applied to arbitrary graphs and depends on both the structure of the graphs and the involved utility functions. The derived sufficient condition can be used during learning in the following way: after the CG is formed, and before max-sum is employed, we evaluate the condition. If it is false, we can decide to avoid coordination for the current iteration and employ independent learning, otherwise, we can proceed with coordination.

We start by formulating the max-sum update equation, initially presented in (6) as:

$$\tilde{\mu}_{I \to i}(a_i) = \max_{a_{\mathcal{N}_I \setminus i}} \big[\ln(u_{\mathcal{N}_I}(a)) + h_{\mathcal{N}_I \setminus i}(a) \big] \tag{44}$$

where we expressed everything in terms of messages from check to variable nodes and defined:

$$h_{\mathcal{N}_I \setminus i}(a) \triangleq \sum_{j \in \mathcal{N}_I \setminus i} \sum_{J \in \mathcal{N}_i \setminus I} \mu_{J \to j}(a_j) \tag{45}$$

To simplify the notation, we denote the utility function $u_{\mathcal{N}_I}(a_{\mathcal{N}_I})$ as $u_{\mathcal{N}_I}(a)$ and messages $h_{\mathcal{N}_I \setminus i}(a_{\mathcal{N}_I \setminus i})$ as $h_{\mathcal{N}_I \setminus i}(a)$. The following theorem is the main tool employed by our analysis:

Theorem 4. (Banach's fixed-point theorem) Let $f : \mathcal{X} \to \mathcal{X}$ be a contraction of a complete metric space $(\mathcal{X}, d)$, where d represents the distance metric. Then, f has a unique fixed point $x^\infty \in \mathcal{X}$ and $\forall x \in \mathcal{X}$, the sequence $x, f(x), f^2(x), \cdots$ obtained by iterating f converges to x^∞. The rate of convergence is at least linear to $d(f(x), x^\infty)$, since $d(f(x), x^\infty) \leq K d(x, x^\infty)$ for all $x \in \mathcal{X}$, where K satisfies $0 \leq K < 1$ and $d(f(x), f(y)) \leq K d(x, y),\ \forall x, y \in \mathcal{X}$.

As suggested by Lemmas 1 and 2 in [12], in order to prove that the message update equation is a contraction, we bound its derivative. Directly taking the derivative of (44) would result in a trivial bound, we thus re-parameterize messages in terms of a monotonically increasing function:

$$\nu_{I \to i}(a_i) = e^{\mu_{I \to i}(a_i)} \tag{46}$$

The derivative of (46) can be calculated as:

$$\frac{\partial \tilde{\nu}_{I \to i}(a_i)}{\partial \nu_{J \to j}(y_j)} = e^{\max_{a_{\mathcal{N}_I \setminus i}} \left(Q_{\mathcal{N}_I}(a) + h_{\mathcal{N}_I \setminus i}(a) \right)} \mathbf{1}_{\mathcal{N}_j \setminus I}(J) \mathbf{1}_{\mathcal{N}_I \setminus i}(j) \tag{47}$$

$$\leq e^{\max_{a_{\mathcal{N}_I \setminus i}} Q_{\mathcal{N}_I}(a) + \max_{a_{\mathcal{N}_I \setminus i}} h_{\mathcal{N}_I \setminus i}(a)} \mathbf{1}_{\mathcal{N}_j \setminus I}(J) \mathbf{1}_{\mathcal{N}_I \setminus i}(j) \tag{48}$$

$$= e^{\max_{a_{\mathcal{N}_I \setminus i}} Q_{\mathcal{N}_I}(a)} e^{\max_{a_{\mathcal{N}_I \setminus i}} h_{\mathcal{N}_I \setminus i}(a)} \mathbf{1}_{\mathcal{N}_j \setminus I}(J) \mathbf{1}_{\mathcal{N}_I \setminus i}(j) \tag{49}$$

We define:

$$A_{I \to i, J \to j} = e^{\max_{x_{\mathcal{N}_I \setminus i}} Q_{\mathcal{N}_I}(a)} \mathbf{1}_{N_j \setminus I}(J) \mathbf{1}_{\mathcal{N}_I \setminus i}(j) \tag{50}$$

$$B_{I \to i}(\nu) = e^{\max_{a_{\mathcal{N}_I \setminus i}} h_{\mathcal{N}_I \setminus i}(a)} \tag{51}$$

Note that we have absorbed all ν-dependence in the term $B_{I \to i}(\nu)$, while term $A_{I \to i, J \to j}$ captures the structure of the bipartite graph, as well as the effect of the utility functions. In order to bound (49), we employ the following theorem:

Theorem 5. (Theorem 2 in [12]) Let $f : \mathbb{R}^m \to \mathbb{R}^m$ be differentiable and suppose that $f'(x) = B(x)A$, where A has nonnegative entries and B is diagonal with bounded entries $|B_{ii}(x)| \leq 1$. If the spectral radius of matrix A is strictly less than 1, then for any $x \in R^m$, the sequence $x, f(x), f^2(x), \cdots$ obtained by iterating f converges to a fixed point x_∞, which does not depend on x.

If we assume that $h_{\mathcal{N}_I \setminus i}(x)$ is normalized in the range (0,1), we can bound $B_{I \to i}(\nu)$ as:

$$\sup |B_{I \to i}(\nu)| \leq e \tag{52}$$

This bound corresponds to a worst-case analysis, where messages from all nodes involved in (50) are vectors with all their elements, except for one, set to 0. In addition, the index of the non-zero element must be the same for all nodes. Although this situation may arise, we expect that the messages exchanged in reality are more uniform. We also anticipate that, the more nodes are involved in (50), the less probable it is that agents agree on the maximum index. In Section VIII-C, we present a heuristic that significantly refines this bound.

If we multiply all elements of matrix $A_{I \to i, J \to j}$ by the bound on the right-hand side of (52) and form matrix $\bar{A}_{I \to i, J \to j}$ with the new elements, then, based on Theorems 4 and 5, we derive the main result of our convergence analysis:

Theorem 6. If the spectral radius of matrix $\bar{A}_{I \to i, J \to j}$ is strictly smaller than 1, then the max-sum algorithm converges to a unique fixed point irrespective of the initial messages. Furthermore, the rate of convergence is at least linear to $d(f(x), x^\infty)$.

VIII. Simulations

A. Simulation Setup

To evaluate the proposed solution, we first examine its performance on a toy network with frames of size $C = 10$ and channel load $G \in [0.1, \cdots, 1]$. We set the buffer size to $B = 3$, the maximum number of replicas d to 8 and the maximum number of max-sum iterations $N_{\max}$ to 10. In each

frame, a sensor accepts a new packet with a probability of 0.5. Regarding the learning parameters, we define a constant exploration rate ϵ of 0.05, a learning rate α of the form $0.9\alpha_b^i$, where α_b is a constant dictating the rate of decay and i denotes the number of visits of the current history-action pair. Furthermore, we employ a constant value for γ. Parameters α_b and γ were tuned for different ranges of G, and we observed that a value of $\gamma = 0.4$ and $\alpha_b = 0.4$ is optimal for $G \leq 0.6$, while $\gamma = 0.98$ and $a_b = 0.98$ works best for high loads. Finally, we employ a fixed window w of 4 observations. Unless stated otherwise, performance is averaged over 1000 Monte Carlo trials. Confidence intervals are calculated based on 20 independent experiments with 97.5% confidence level. To prove the superiority of our solution to traditional MAC, we compare its performance with an IRSA protocol where $\Lambda(x) = 0.25x^2 + 0.60x^3 + 0.15x^8$, which proved superior to other commonly used distributions [3] as well as with our previous method, presented in [4].

B. Throughput evaluation

The purpose of our evaluation is twofold. First, we aim to prove that learning-based protocols surpass in performance the current state-of-the-art random access protocol, IRSA. Second, we want to draw insights into how coordinated learning compares with independent and centralized approaches. In Fig. 4, we observe that all learning-based methods improve upon IRSA in terms of the normalized throughput, and, thus, confirm the necessity of adaptive solutions. In addition, we observe that the throughput achieved by the proposed, coordinated approach, is higher than the independent learning case, and lower than the centralized solution for $G < 0.8$. This result was anticipated, as agents that coordinate their actions using the max-sum algorithm converge to neighbourhood optima, in contrast to independent agents that get stuck in local optima. In contrast, a centralized approach solves the problem in the original, joint space, Q-learning can thus converge to the global optimum. Note that, due to memory restrictions, we were not able to apply the centralized solution on networks with frame size larger than $D = 5$.

Fig. 5 exhibits the benefits of coordination in terms of convergence rate and achieved throughput. We observe that coordinating agents converge early to higher throughput than independent agents, which converge slowly and experience oscillations. Although convergence of multi-agent reinforcement learning algorithms is a largely unexplored area, Fig. 5 is in accordance with our expectations: when sensors act independently, they learn in an uncertain environment and, thus, require more learning iterations. In contrast, agents that coordinate learn more steadily and converge to a well performing solution within a few iterations.

C. Robustness evaluation

We evaluate robustness of learning based on the sufficient condition presented in Section VII. In Fig. 6, we present the probability of convergence of the max-sum algorithm, measured as the percentage of times that the condition indicated a possible failure to converge ($\|\bar{A}_{I\to i,J\to j}\| > 1$). We observe

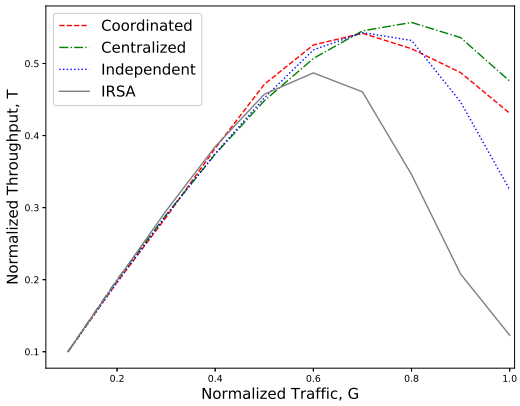


Fig. 4: Achieved throughput comparison of vanilla IRSA and three learning-based IRSA designs.

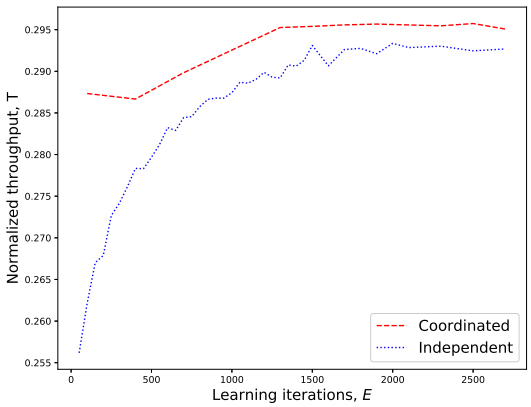


Fig. 5: Convergence rate of Coordinated and Independent agents for channel traffic $G = 0.3$.

that convergence is less likely to be guaranteed for higher G. Additionally, this figure presents how convergence varies with the Q-table initialization: the values of all entries of the local Q-tables of sensors are initialized by randomly sampling in the interval $[-B - c, -B + c]$, where B is the capacity of the buffer of agents and c is a constant that was chosen to have very low ($c = 0.01$) or very high ($c = 4$) value. Note that this initialization of the Q-tables corresponds to a uniform distribution with mean equal to B and variance given by $4c^2/12$. We observe that higher randomization weakens the convergence guarantees.

In order to gain further insights into how convergence depends on the CG realizations encountered during learning, we separately evaluate the spectral radius of matrix $A_{I\to i,J\to j}$ and the bound of matrix $B_{I\to i}$. In Fig. 7, we present how the spectral radius of $A_{I\to i,J\to j}$ evolves with learning iterations. In particular, we calculate its moving average for a window of 70. We observe that the spectral radius in general increases with learning time for low ($G = 0.2$), intermediate ($G = 0.5$) and high ($G = 0.7$) channel loads. Also, it is significantly higher for intermediate channel loads. Both these observations

can be justified by closely examining (50): matrix $A_{I\to i,J\to j}$ has mostly zero entries, denoting the absence of collisions. Also, the number of collisions tends to decrease as the learning process proceeds, due to agents learning how to avoid each other, and increase with the channel load. However, the non-zero entries acquire higher values as the learning process proceeds, due to agents becoming more certain of which actions are optimal. In addition, collisions are more common for high G, thus Q-tables, as well as the matrix $A_{I\to i,J\to j}$, have lower entry values, as collisions likely lead to failure to transmit and thus, lower rewards. Note that this is only evident for high loads ($G = 0.7$). As indicated in Fig. 4, throughput is optimal for both $G = 0.2$ and $G = 0.5$, the lower values for $G = 0.2$ can be, therefore, attributed to the higher sparsity of the coordination graph.

In Fig. 8, we evaluate how the bound of $B_{I\to i}$ differs from the worst-case scenario, based on a heuristic evaluation. In particular, our simulations indicate that sensors tend to send different number of replicas for different frames for medium channel load ($G \in [0.4, 0.5, 0.6]$), in order to efficiently exploit the available slots, while they all agree to send a few replicas (1 or 2) when the channel load is high ($G = 0.7$). Based on these observations, we evaluate the bound of (52) using the current messages of check nodes, and observe that it takes significantly lower values than the worst-case analysis. It is worth noting that this heuristic evaluation gives lower values for $G = 0.7$ than for $G = 0.5$, as, with increasing sizes of the Q-tables, it is less likely that all agents associated with a Q-table will agree on a common action.

D. Complexity

Fig. 9 exhibits the benefits of our complexity reduction technique, which we measure as the average number of agents' collisions. We observe that throughput is low when complexity reduction is not employed, as the learning algorithm exhausts the time budget before a good policy is found. We can, thus, conclude that the reduction in complexity achieved is particularly significant for high channel loads ($G \geq 0.6$), where collisions are frequent and lead to CGs of low sparsity. Finally, in Fig. 10, we present the time complexity of the different techniques employed in our solution. Note that, in these simulations, learning was rendered robust by employing the sufficient condition derived in Section VII, we therefore avoid cases that would require exhausting the number of message-passing iterations. Furthermore, by employing the MKP complexity reduction technique, CGs become more sparse and, therefore, significantly reduced time is required for coordination. It is, thus, anticipated, that the computation time of max-sum would have been much higher, had these two techniques not been employed. We observe that calculating the condition for robustness is the main bottleneck of the operation and time complexity increases significantly with the channel load G.

Finally, we perform an experiment that examines how our solution performs for different WSN topologies. Fig. 11 presents the throughput and time required for learning for two types of networks: a fully-connected (simple) WSN with

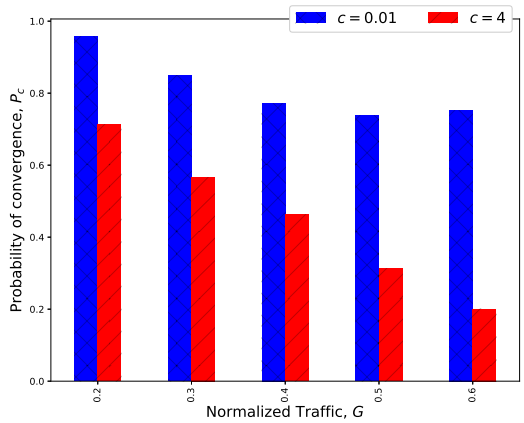


Fig. 6: Evaluation of the sufficient condition for robustness for different channel loads and different initialization for the local Q-tables.

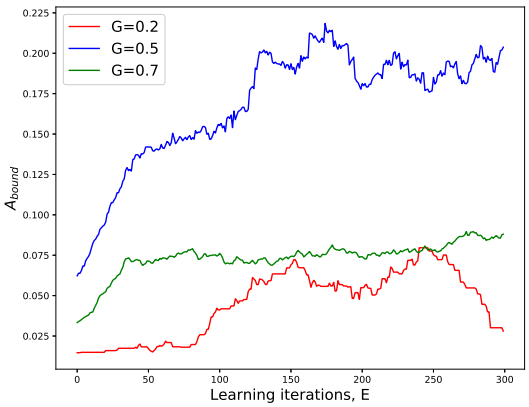


Fig. 7: The evolution of spectral radius of matrix A with the learning iterations for varying G.

$C = 16$ and $D = 20$, and a WSN with sensors clustered into 4 clusters of the same size. We assume that sensors can collide only with sensors within their cluster. From Fig. 11, we observe that, in a fully-connected network, learning is significantly slower, as the time budget is exhausted at 100 learning iterations. This suggests that the complexity of coordination is high due to CGs not being sparse enough. Regarding throughput, the clustered network achieves significantly higher performance, which can be attributed to: (i) experiencing more learning iterations in the same time period, and, (ii) coordination being more important, as clustering dependencies are persisting, whereas dependencies that arise due to collisions may change at each learning iteration. We base this conclusion on the observation that learning in the fully-connected network achieves lower throughput, even when the time budget is not exhausted ($L_E < 100$). Note that the complexity of our solution does not depend on the number of clusters, but increases with the size of the clusters. Due to the locality of interaction in WSNs, it is natural to decompose the network into clusters that operate independently from each other. The

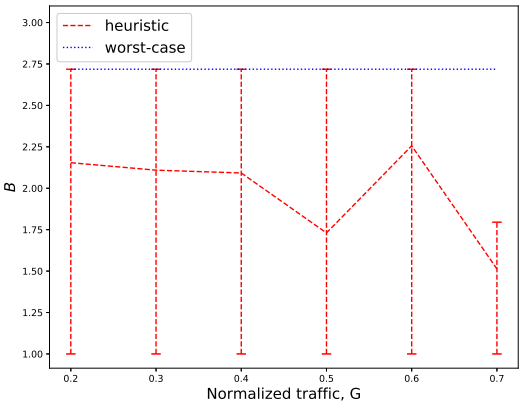


Fig. 8: Evaluation of the bound of matrix B for different channel loads based on the worst-case theoretical analysis and heuristically calculated based on the values of the messages during simulation.

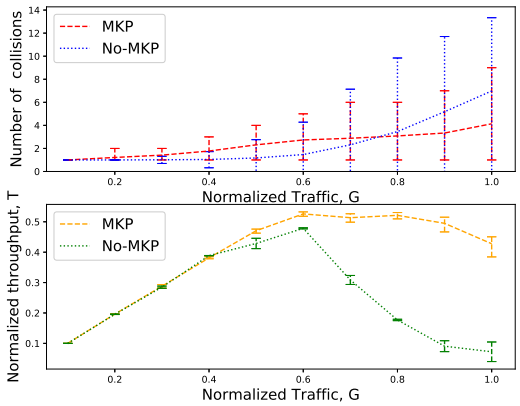


Fig. 9: Number of collisions sensors experience upon transmission and achieved throughput for Coordinated agents before and after reducing complexity using the MKP technique.

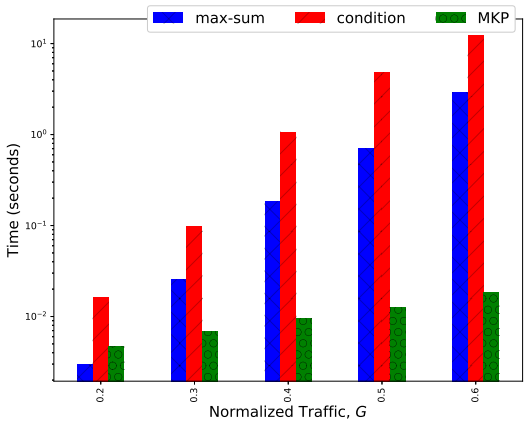


Fig. 10: Time (measured in seconds) for the different stages of learning: application of max-sum, calculation of robustness condition and solution of the MKP.

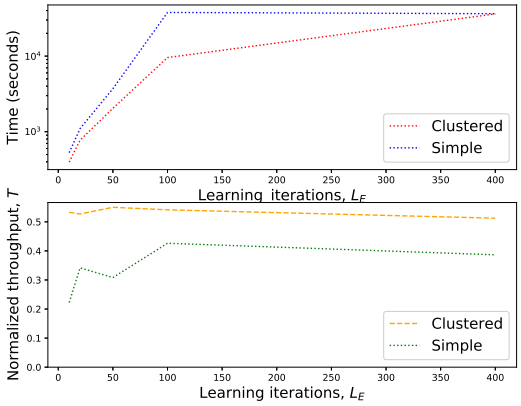


Fig. 11: Comparison of time complexity and achieved average throughput for a fully-connected network (simple) with 16 sensors, and a clustered one with 4 group of sensors with 4 sensors each for $G = 0.8$

sizes of these clusters should remain small compared to the size of the network and their value is dictated by practical limitations of sensors, such as their transmission power. We also simulated networks consisting of clusters of 4 sensors, channel load $G = 0.8$ and $C \in \{160, 320, 480\}$, and confirmed that the achieved throughput remained approximately 0.56 in all cases.

IX. Conclusions and Future Work

In this paper, we presented a MAC solution for WSNs based on coordinated reinforcement learning, where the max-sum algorithm is applied on CGs, used to describe the dependencies among sensors, to find the optimal actions. We derived a technique for bounding complexity based on a multiple knapsack formulation, as well as convergence guarantees for our Q-learning based algorithm in our framework, which we termed as Groupwise-Dependent Decentralized POMDP. Our simulations confirm that coordination is beneficial in terms of throughput and convergence rate, when compared with classical MAC, as well as solutions employing independent learning. Furthermore, coordination must exploit structural properties, as well as complexity reduction techniques, so that computational complexity does not prohibit learning. Our current solution implicitly promotes energy efficiency by reducing the complexity of coordination and ensuring robustness of the learning process. As part of future work, we plan to investigate how energy efficiency can be explicitly considered in our framework. This will necessitate adjusting the modeling of the POMDP and the optimization objective to directly minimize energy consumption or maximize the lifetime of the WSN. Also, weights and capacities of our MKP will need to be reformulated to account for other typical constraints in WSNs, such as the battery lifetime of sensors.

REFERENCES

[1] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *arXiv:1809.08707 [cs]*, Sept. 2018. [Online]. Available: http://arxiv.org/abs/1809.08707

[2] N. M. Abramson, "THE ALOHA SYSTEM: Another Alternative for Computer Communications," in *Proc. of joint Computing Conf. AFIPS'70*, Honolulu, HI, USA, Nov. 1970.

[3] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted aloha," *IEEE Trans. on Communications*, vol. 59, no. 2, pp. 477–487, Feb. 2011.

[4] E. Nisioti and N. Thomos, "Decentralized reinforcement learning based mac optimization," in *IEEE Proc. of PIMRC 2018, Bologna, Italy*, Sep. 2018.

[5] C. Guestrin, D. Koller, and R. Parr, "Multiagent planning with factored mdps," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2002, pp. 1523–1530.

[6] C. Claus and G. Boutilier, "The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems," in *Proc. of the AAAI '98/IAAI '98*, Madison, WI, USA, Jul. 1998.

[7] C. Zhang and V. Lesser, "Coordinated multi-agent reinforcement learning in networked distributed pomdps," in *Proc. of the 25th Conf. on Artificial Intelligence, AAAI'11*, San Francisco, CA, USA, Aug. 2011.

[8] C. Guestrin, M. G. Lagoudakis, and R. Parr, "Coordinated reinforcement learning," in *Proc. ICML '02*, San Francisco, CA, USA, Jul. 2002, pp. 227–234.

[9] Y.-M. D. Hauwere, P. Vrancx, and A. Now, "Generalized learning automata for multi-agent reinforcement learning," *IOS Press Journal of AI Communications*, vol. 23, no. 4, pp. 311–324, Apr. 2010.

[10] M. Paskin, C. Guestrin, and J. McFadden, " A Robust Architecture for Distributed Inference in Sensor Networks," in *Proc. IPSN '05*, Apr. 2005.

[11] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990.

[12] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of the sumproduct algorithm," *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4422–4437, Dec. 2007.

[13] G. L. Choudhury and S. S. Rappaport, " Diversity ALOHA A Random Access Scheme for Satellite Communications," *IEEE Trans. on Communications*, vol. 31, no. 3, pp. 450–457, Mar. 1983.

[14] E. Casini, R. D. Gaudenzi, and O. D. R. Herrero, " Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Schemefor Satellite Access Packet Networks," *IEEE Trans. on Wireless Communications*, vol. 6, no. 4, pp. 1408–1419, Apr. 2007.

[15] L. Toni and P. Frossard, "IRSA transmission optimization via online learning," *CoRR*, vol. abs/1801.09060, 2018. [Online]. Available: http://arxiv.org/abs/1801.09060

[16] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *arXiv:1704.02613 [cs]*, 2017. [Online]. Available: http://arxiv.org/abs/1704.02613

[17] U. Challita, L. Dong, and W. Saad, "Proactive resource management in LTE-u systems: A deep learning perspective," *arXiv:1702.07031 [cs, math]*, 2017. [Online]. Available: http://arxiv.org/abs/1702.07031

[18] J. R. Kok and N. Vlassis, "Sparse Cooperative Q-learning," in *Proc. ICML*, Jul. 2004, pp. 61–.

[19] ——, "Using the max-plus algorithm for multiagent decision making in coordination graphs," in *RoboCup 2005: Robot Soccer World Cup IX*, 2006, vol. 4020, pp. 1–12.

[20] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," *CoRR*, vol. abs/1301.6725, 2013. [Online]. Available: http://arxiv.org/abs/1301.6725

[21] A. Rogers, A. Farinelli, R. Stranders, and N. Jennings, "Bounded approximate decentralised coordination via the max-sum algorithm," *Artificial Intelligence*, vol. 175, no. 2, pp. 730–759, Feb. 2011.

[22] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Computation*, vol. 12, no. 1, pp. 1–41, Jan. 2000.

[23] J. R. Kok and N. Vlassis, " Collaborative Multiagent Reinforcement Learning by Payoff Propagation," *J. Mach. Learn. Res.*, vol. 7, pp. 1789–1828, Dec. 2006.

[24] Y. Weiss and W. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 736–744, Feb. 2001.

[25] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1-2, pp. 99–134, May 1998.

[26] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

[27] J. L. R. Ford and D. R. Fulkerson, "A suggested computation for maximal multi-commodity network flows," *Management Science*, vol. 5, no. 1, pp. 97–101, 1958.

[28] J. M. Bernardo and A. F. Smith, *Bayesian Theory*. John Willen & Sons, Inc., 1994.

[29] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1, pp. 1–305, 2007.

[30] G. T. Ross and R. M. Soland, "A branch and bound algorithm for the generalized assignment problem," *Mathematical Programming*, vol. 8, no. 1, pp. 91–103, Dec 1975.

Eleni Nisioti is a PhD candidate at the University of Essex, UK at the Department of Computer Science and Electronics Engineering. She is part of the Communications and Networks group and her research interests revolve around the application of machine learning for communications with a focus on multi-agent reinforcement learning for wireless sensor networks. She received the Diploma in Electrical and Computer Engineering degree from Aristotle University of Thessaloniki in 2017.

Nikolaos Thomos (S'02-M'06-SM'16) is an Associate Professor at the University of Essex, UK and the group leader of the Communications and Networks group. Previously, he was senior researcher at the Ecole Polytechnique Fédérale de Lausanne (EPFL), and the University of Bern, Switzerland. He received the highly esteemed Ambizione career award from Swiss National Science Foundation (SNSF). He received the Diploma and Ph.D. degrees from Aristotle University of Thessaloniki, Greece in 2000 and 2005 respectively. He is an elected member of IEEE MMCP Technical Committee (MMSP - TC) for the period 2019 - 2022. His research interests include machine learning for communications, multimedia communications, network coding, information-centric networking, source and channel coding, device-to-device communication, and signal processing.