IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 28, 2019, accepted February 22, 2019, date of publication February 28, 2019, date of current version March 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902358

An Optimal Greedy Algorithm for the Single Access Contention Resolution Problem

ITZEL C. OLIVOS-CASTILLO[1], RICARDO MENCHACA-MENDEZ[1], ROLANDO MENCHACA-MENDEZ[1], MARCELO M. CARVALHO[2], (Member, IEEE), AND MARIO E. RIVERO-ANGELES[1], (Member, IEEE)

[1]Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City 07738, Mexico
[2]University of Brasília, Brasília 70919-970, Brazil

Corresponding author: Ricardo Menchaca-Mendez (ric@cic.ipn.mx)

This work was supported in part by the CONACyT, and in part by the Instituto Politécnico Nacional.

ABSTRACT We present the greedy optimal algorithm for contention resolution (GOAL-CR), a greedy algorithm that solves a variant of the standard contention resolution problem where a set of nodes want to access a shared resource only once, and the objective is to minimize the time it takes for all the nodes to access the resource successfully. These assumptions hold, for instance, in event reporting applications or in the cluster formation phase of wireless sensor networks. We formally prove that the GOAL-CR computes access policies that minimize the expected contention resolution time. We also show, numerically, that the performance of the greedy policies is close to that of a protocol with complete information about the exact number of nodes that have not yet accessed the resource; this latter assumption is hard to fulfill in practice but allows the derivation of a lower bound for the problem. In addition, we show how to adapt the algorithm to scenarios where there is uncertainty in the initial number of nodes and to scenarios where nodes have very limited memory. Finally, we use simulations to show the robustness of the GOAL-CR against asynchronous starts.

INDEX TERMS Algorithms, contention resolution, greedy algorithms, optimization, random algorithms, wireless sensor networks.

I. INTRODUCTION

The standard contention resolution problem consists of devising a protocol that allows a set of n nodes to access a shared resource when there is no form of communication among them. The synchronous version of the problem assumes that time is divided into discrete time slots and that the resource can be accessed by at most one node in every single slot, that is, if two or more entities attempt to access simultaneously they lock each other out, and none can access the resource during the slot. This problem occurs, for example, in distributed systems where a set of n processes want to access a shared database or in wireless networks where nodes share a communication channel.

The most common solution for the contention resolution problem uses randomization as follows: during each slot, each of the nodes tries to access the resource with probability $p > 0$ independently of the other nodes. The value of $p = 1/n$ is the one that maximizes the probability that exactly one, out of n nodes, tries to access the resource [1]. In this paper, we solve a variant of the standard contention resolution problem in which the nodes need to access the shared resource only once, and the goal is to minimize the expected value of the number of slots needed for all nodes to access the resource successfully, hereafter defined as contention resolution time. Since the number of nodes decreases each time a successful access occurs, it is suboptimal to use a fixed probability p during the whole duration of the contention. Instead, nodes should try to access the resource at each slot t with probability $p_t = 1/A_t$ where A_t is the number of active nodes at slot t. However, because nodes cannot communicate among themselves, they cannot know the current number of active nodes A_t. In this work, we address the problem of computing the sequence of probabilities $\{p_t\}$, which we will refer as policy, that minimizes the expected value of the contention resolution time without assuming that the nodes know, or can estimate, the value A_t.

The primary motivating application for this variant of the problem is the cluster formation phase of routing protocols for wireless sensor networks (WSNs). Routing in a WSN is

The associate editor coordinating the review of this manuscript and approving it for publication was Shagufta Henna.

different from other wireless networks because sensors have energy constraints and they can be randomly deployed. For such scenarios, protocols like LEACH [2] are implemented to periodically produce strategic clusters that allow to establish hierarchical routing schemes and to balance the sensors' energy consumption. Each time the clusters are reorganized, the protocols run a set-up phase in which each sensor sends a single packet through a broadcast communication channel with information about its current location and energy level. Minimizing the contention resolution time of the set-up phases not only impacts metrics like channel utilization, throughput, and energy consumption but also impacts critical metrics like the response time of a WSN to a catastrophe like a wildfire or an earthquake [3].

In the context of WSNs, it is common that nodes attempt transmissions following a CSMA protocol with a fixed transmission probability p. The value of p is appropriately selected at the beginning of the network operation for some particular condition, and then, the nodes use it all along the random access [4]; in [5], for example, the nodes attempt transmissions with the value that maximizes their chances of successfully transmitting a message within a delivery deadline. Unfortunately, although the use of a fixed optimal value of p saves energy by avoiding extra computation in the nodes, it entails a sub-optimal average contention resolution time of $O(n \log n)$ and $\omega(n^{1-\epsilon} \log n)$ slots for any $\epsilon > 0$ (see Appendix B-2). On the other hand, adaptive strategies that maximize channel utilization by tuning the value of p at run-time, e.g., [6]–[8], entail an increased cost regarding energy utilization since they require the nodes to estimate the network traffic conditions continuously. In [6], the nodes sense the channel to measure the average number of idle slots and the average number of slots that resulted in a collision. Backoff approaches like [7] also require to keep track of past trials of transmission. Bruno *et al.* [8] show that the optimal capacity of p-persistent CSMA is achieved when $p = 1/A_t$ and point out that, in general, it is difficult to have a precise knowledge of the number of active nodes. An alternative solution consists in continuously computing the current number of active nodes at the base station and then broadcasting the optimal value of p; however, this information would have to be reliably transmitted to the remaining sensors, which increases delay and reduces throughput. Therefore, new approaches must be developed to approximate the optimal contention resolution time without neither increasing energy consumption nor using extra bandwidth.

A. PROPOSED SOLUTION

In this paper, we introduce the Greedy Optimal Algorithm for Contention Resolution (GOAL-CR). GOAL-CR attempts to provide the advantages of both fixed and adaptive schemes while reducing their respective disadvantages; to achieve this, our algorithm considers the probable evolution of the contention and computes, *offline*, the appropriate sequence of access probabilities. In this way, the nodes do not need to waste energy computing the right access probabilities during the run-time of the contention, and still use access probabilities that result in contention resolution times close to the optimal.

More specifically, GOAL-CR maintains a probability distribution of the number of active nodes and, for each time slot t, chooses as access probability the one that maximizes the likelihood of a successful resource acquisition given the current distribution of active nodes. We prove that the computed sequences minimize the expected contention resolution time. Moreover, we show, using simulations, that GOAL-CR achieves contention resolution times close to the resolution times of a protocol with perfect information, where nodes know the exact number of active nodes A_t at all times. The perfect-information protocol has expected resolution time of order $\Theta(n)$ (see Appendix B-1). In this sense, GOAL-CR performs similarly to the best possible protocol.

In addition to minimizing the contention resolution time, GOAL-CR has another useful property. It can be used in scenarios where the exact number of nodes that will participate in the contention is not known for sure. For example, when a group of sensors is deployed from an aircraft to track the dynamics of a wildfire, some of them may fail. In such a case, a probabilistic model of the number of surviving sensors can be proposed and incorporated into the algorithm as the initial distribution of active nodes.

II. GOAL-CR: GREEDY OPTIMAL ALGORITHM FOR CONTENTION RESOLUTION

As we mentioned, the objective of GOAL-CR is to build a sequence of access probabilities (or policy) that minimizes the expected contention resolution time. In order to use combinatorial optimization methods, we constrained the available access probabilities p_t to be in a finite set B. This discretization is similar to how actions and states are discretized when one wants to use reinforcement learning in continuous environments [9]. The resulting discrete problem can be modeled as a partially observable Markov decision process (POMDP) [10] where the partial observability is in the uncertainty about the number of active nodes and the decisions are the probabilities chosen at each step. Solving a general POMDP is computationally hard (in [11] this problem is shown to be PSPACE-complete), however, in this paper we show that the specific instances that result from minimizing the expected contention resolution time can be solved in polynomial time using GOAL-CR.

GOAL-CR is a greedy algorithm that at each step selects the access probabilities that minimize the expected value of the number of active nodes at slot $t+1$, given the distribution (or belief) of active nodes at slot t. This is equivalent to maximizing the probability of having a successful access at slot t (see Proposition 1 of Appendix A). More precisely, let $\pi = (g_1, \ldots, g_T)$ denote a particular policy computed by GOAL-CR, and let A_t^π denote the random variable that counts the number of active nodes at slot t when following policy π. GOAL-CR (see Algorithm 1) receives an initial belief $\mathbf{a}[\cdot]$ with values $\mathbf{P}(A_{t=0}^\pi = i)$, i.e., a vector denoting the probability

Algorithm 1 GOAL-CR

Input: $\mathbf{a}$ = initial belief of active nodes, T = length of the policy, B = finite set of available access probabilities.

1: **for** $t = 1$ to T **do**
2: $\quad g_t = argmax_{q\in B} \sum_i iq(1-q)^{i-1}\mathbf{a}[i]$
3: $\quad$ **for all** i **do**
4: $\quad\quad failure = (1 - ig_t(1-g_t)^{i-1})\mathbf{a}[i]$
5: $\quad\quad success = (i+1)g_t(1-g_t)^{i}\mathbf{a}[i+1]$
6: $\quad\quad \mathbf{a}'[i] = success + failure$
7: $\quad$ **end for**
8: $\quad \mathbf{a}[\cdot] \leftarrow \mathbf{a}'[\cdot]$
9: **end for**

Output: $\pi = \{g_1, g_2, \cdots, g_T\}$

Algorithm 2 Continuous GOAL-CR

Input: $\mathbf{a}$ = initial belief of active nodes, T = length of the policy.

1: **for** $t = 1$ to T **do**
2: $\quad g_t = argmax_{q\in[0,1]} \sum_i iq(1-q)^{i-1}\mathbf{a}[i]$
3: $\quad$ **for all** i **do**
4: $\quad\quad failure = (1 - ig_t(1-g_t)^{i-1})\mathbf{a}[i]$
5: $\quad\quad success = (i+1)g_t(1-g_t)^{i}\mathbf{a}[i+1]$
6: $\quad\quad \mathbf{a}'[i] = success + failure$
7: $\quad$ **end for**
8: $\quad \mathbf{a}[\cdot] \leftarrow \mathbf{a}'[\cdot]$
9: **end for**

Output: $\pi = \{g_1, g_2, \cdots, g_T\}$

of having i active nodes at the beginning of the network operation. Then, the purpose of each iteration t of the outer for-loop is to add a new access probability to policy π and to maintain an updated belief of active nodes. Accordingly, GOAL-CR chooses the probability g_t from B that maximizes the chances of having a successful resource acquisition $\big($i.e., $\sum_i iq(1-q)^{i-1}\mathbf{P}(A_t^\pi = i)$, line 2$\big)$ and updates the belief of active nodes based on its current choice. The belief-updating procedure is computed in the inner for-loop using the observation that A_{t+1}^π takes a particular value i with probability:

$$\mathbf{P}(A_{t+1}^\pi = i) = \mathbf{P}(A_{t+1}^\pi = i | A_t^\pi = i)\mathbf{P}(A_t^\pi = i) + \mathbf{P}(A_{t+1}^\pi = i | A_t^\pi = i+1)\mathbf{P}(A_t^\pi = i+1)$$

where $\mathbf{P}(A_{t+1}^\pi = i | A_t^\pi = i) = 1 - ig_t(1-g_t)^{i-1}$ (line 4) and $\mathbf{P}(A_{t+1}^\pi = i | A_t^\pi = i+1) = (i+1)g_t(1-g_t)^{i}$ (line 5) are the probabilities of having a failed or a successful access respectively.

In Theorem 1 (see Section III), we show that our algorithm gives a policy that yields the minimum expected number of active nodes for all time slots t among all policies that only use probabilities in the set B. Therefore, we can think of GOAL-CR as an algorithm that produces both, a rigorous upper bound on the (global) minimum expected number of active nodes for any time slot t and set B, and a policy that achieves such upper bound. To get a reasonable upper bound, we included in B the values of the probabilities that nodes would use if they knew the exact number of active nodes at each time slot, i.e., $B = \{1/n, 1/(n-1), \ldots, 1/2, 1\}$. Our simulation results show that this choice of B gives results very close to the performance of a perfect-information protocol that knows the exact number of active nodes at all time slots.

Regarding the input parameter T, we chose a sufficiently large value so that the expected number of active nodes at time slot T is close to zero. Our empirical evidence shows that $\sum_i i\mathbf{P}(A_T^\pi = i) \approx 0$ when $T \approx 3\cdot n$. This motivates us to formally show, in future works, that the expected contention resolution time of GOAL-CR is $O(n)$.

To illustrate the behavior of the proposed algorithm, in Fig. 1 we show the evolution of the distribution of active nodes A_t^π when using a greedy policy and an optimal fixed probability to solve the contention in a network of size $n = 200$. Observe that the probability mass moves from n to 0 in both cases, but moves faster when using the greedy policy. Also, notice that the distribution of active nodes tends to be concentrated around its mean, this property increases the chance of selecting the right access probability at each time slot and explains why the performance of the greedy policies is close to the performance of the perfect-information protocol.

The running time of GOAL-CR is $O(T\cdot(n\cdot b+n))$ where b is the cardinality of B. Thus, our choice of $B = \{1/n, 1/(n-1), \ldots, 1/2, 1\}$ results in a running time of $O(T\cdot n^2)$. We could obtain slightly better bounds if we increased the granularity of the set B but at the cost of increasing the running time. The final granularity of the set B will depend on the computational resources at hand. As an example, consider the scenario where a WSN is deployed, and the sensors try to inform a base station of its existence. The policies could either: *i*) be computed by a node designated as base station and then broadcasted to the remaining nodes; or *ii*) be computed prior the network deployment by a powerful computer and then stored in the memory of the nodes. In the former case, the computational capacity, memory, and energy are very restricted and hence, having a low complexity algorithm is of paramount importance. In the latter case, the plentiful of computational resources allows to trade more complexity for performance gains.

A. CONTINUOUS GOAL-CR

In this section, we introduce a modification of GOAL-CR (see Algorithm 2) where we replaced the optimization problem of line 2 of Algorithm 1 with a continuous optimization problem, i.e., instead of optimizing over the discrete set B, continuous GOAL-CR optimizes over the whole interval [0, 1], namely

$$g_t = argmax_{q\in[0,1]} \sum_{i=0}^{n} iq(1-q)^{i-1}\mathbf{a}[i] \tag{1}$$

In Section III, we argue that the proof of correctness of the basic algorithm still applies in the continuous case as

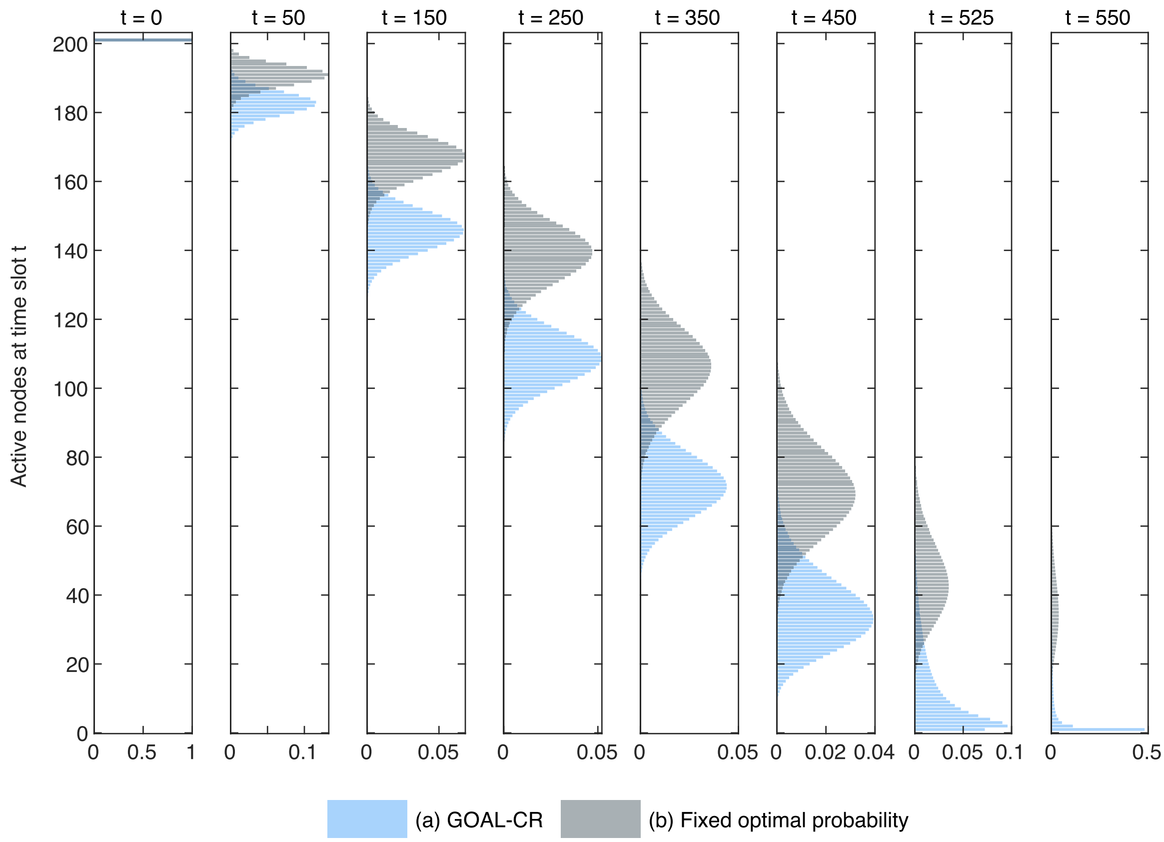

FIGURE 1. Evolution of the distribution of active nodes given a network of size $n = 200$. During the run-time of the contention, the nodes either (a) adjust their access probability following a policy computed by GOAL-CR or (b) attempt to access the resource with a fixed optimal probability computed at the beginning of the network operation.

long as we have a method to solve the above optimization problem. Fortunately, the function defined in (1) is a polynomial of degree $n + 1$ of a single variable q. Thus, to find the value of q that maximizes the likelihood of a successful resource acquisition, we just need to compute all the real roots in [0, 1] of the polynomial of degree n that results from differentiating (1) and picking the one that achieves the maximum. The problem of finding all the real roots of a polynomial of degree n on a given interval is a fundamental task in computer algebra, and many methods have been developed to solve it. The best-known algorithm has bit complexity of $O(n^3 + rn^2)$ where r is the number of bits needed to represent the coefficients of the polynomial [12]. Hence, the overall complexity of continuous GOAL-CR is $O(T \cdot n^3 + T \cdot rn^2)$, which is an order of magnitude worse than the basic GOAL-CR. Nevertheless, this continuous version computes global optimal policies, i.e., policies that minimize the average contention resolution time over all possible policies and not just over the policies where each access probability is restricted to be in the set B. Besides, although in the worst case we need to pay an extra order of magnitude to compute global optimal policies, in practice, the optimization problem is usually a unimodal function of q. As an example, observe the plots in Fig. 2 that correspond to the optimization problem (1) at different times of the execution of GOAL-CR given a network of size $n = 100$. Due to this property, a fast optimization algorithm that simply finds local optima, like quasi-newton, would suffice to compute global optimal policies.

III. ANALYSIS

The main objective of the proof of correctness of GOAL-CR is to formalize the intuitive notion that maximizing the probability that exactly one node tries to access the resource at each slot is always the correct decision if one wants to minimize the expected contention resolution time. Or alternatively, that it is not possible to increase the likelihood of having successful accesses in future slots by not using the probability that maximizes the chances of having a successful access at the current slot. The analysis is organized as follows. First, we show in Proposition 1 (see Appendix A) that given the belief of active nodes at slot t, minimizing the expected number of active nodes at slot $t+1$ is equivalent to maximizing the probability of having a successful access at slot t. Next, in Proposition 2 (see Appendix A) we show that, independently of the policy, the difference between the expected values of the number of successful accesses, when the number of initial nodes differs by one, is small (less than one). Lastly, in Theorem 1 we combine both results with an exchange argument to show the optimality of the greedy algorithm.

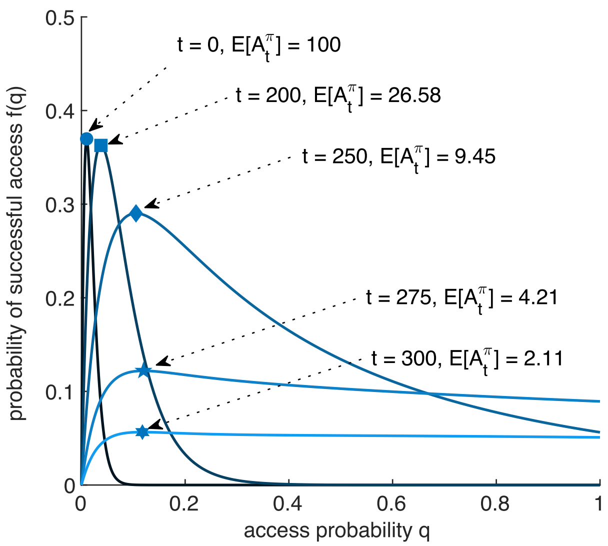

FIGURE 2. Plots of the optimization problem (1) at different times t of the execution of GOAL-CR, given a network of size $n = 100$.

Theorem 1: For any initial distribution of active nodes, GOAL-CR computes a sequence of access probabilities that minimize the expected value of the number of active nodes for all time slots t.

Proof: Consider an optimal sequence of access probabilities $\pi_0 = (p_1, ..., p_t)$ that produces the minimum expected number of active nodes at slot t. We are going to show that it is possible to gradually transform π_0, preserving its optimality at each step, into a sequence where each access probability has been selected according to the greedy rule.

Let $r \leq t$ be the first time slot when policy π_0 does not use the greedy rule, i.e., the access probability p_r is not equal to $g_r = \text{argmax}_{q\in B} \sum_i iq(1-q)^{i-1}\mathbf{P}(A_r^{\pi_0} = i)$. Let π_1 be the same policy as π_0 but with p_r replaced with g_r. We show that $\mathbf{E}[A_t^{\pi_1}] \leq \mathbf{E}[A_t^{\pi_0}]$.

Let W_t^π be a random variable that indicates the number of successful access up to slot t under policy π. We can express the number of active nodes at slot t under policy π as $A_t^\pi = n - W_t^\pi$ and therefore, minimizing $\mathbf{E}[A_t^\pi]$ is equivalent to maximizing $\mathbf{E}[W_t^\pi]$. We write $W_t^\pi = X^\pi + Z^\pi + Y^\pi$ where X^π, Z^π and Y^π count the number of successful accesses in the interval from 0 to $r-1$, the slot r, and the interval from $r+1$ to t respectively. The expected value of W_t^π is $\mathbf{E}[W_t^\pi] = \mathbf{E}[X^\pi] + \mathbf{E}[Z^\pi] + \mathbf{E}[Y^\pi]$. Using the following two observations we show that $\mathbf{E}[W_t^{\pi_1}] - \mathbf{E}[W_t^{\pi_0}] \geq 0$.

The first observation is that $\mathbf{E}[X^{\pi_0}] = \mathbf{E}[X^{\pi_1}]$ since both policies are identical up to slot $r-1$.

The second observation is that $\mathbf{E}[Z^\pi]$ is equal to the probability of a successful access at slot r; hence, $\mathbf{E}[Z^{\pi_1}] \geq \mathbf{E}[Z^{\pi_0}]$ because π_1 is using the greedy rule.

Let S_r^π denote an indicator r. v. of a successful access at slot r under policy π. Let $\mathbf{P}[S_r^\pi]$ and $\mathbf{P}[F_r^\pi]$ denote the probabilities $\mathbf{P}[S_r^\pi = 1]$ and $\mathbf{P}[S_r^\pi = 0]$ respectively. We use the law of total expectation to obtain:

$$\begin{aligned}\mathbf{E}[Y^\pi] &= \mathbf{P}[S_r^\pi]\mathbf{E}[Y^\pi|S_r^\pi] + (1-\mathbf{P}[S_r^\pi])\mathbf{E}[Y^\pi|F_r^\pi] \\ &= \mathbf{E}[Y^\pi|F_r^\pi] + \mathbf{P}[S_r^\pi](\mathbf{E}[Y^\pi|S_r^\pi] - \mathbf{E}[Y^\pi|F_r^\pi]).\end{aligned}$$

We claim that $\mathbf{E}[Y^{\pi_0}|S_r^{\pi_0}] = \mathbf{E}[Y^{\pi_1}|S_r^{\pi_1}]$ because, up to this point, both policies only differ at slot r, more precisely,

$$\begin{aligned}&\mathbf{E}[Y^{\pi_0}|S_r^{\pi_0}] \\ &= \sum_{i=0}^{n} \mathbf{P}(A_r^{\pi_0} = i)\,\mathbf{E}[Y^{\pi_0}|S_r^{\pi_0}, A_r^{\pi_0} = i] \\ &= \sum_{i=0}^{n} \mathbf{P}(A_r^{\pi_1} = i)\,\mathbf{E}[Y^{\pi_0}|A_{r+1}^{\pi_0} = i-1] \\ &= \sum_{i=0}^{n} \mathbf{P}(A_r^{\pi_1} = i)\,\mathbf{E}[Y^{\pi_1}|A_{r+1}^{\pi_1} = i-1] = \mathbf{E}[Y^{\pi_1}|S_r^{\pi_1}],\end{aligned}$$

where in the second to last step we used the fact that $\mathbf{P}(A_r^{\pi_1} = i) = \mathbf{P}(A_r^{\pi_0} = i)$, since both policies are the same before slot r, and that the equality $\mathbf{E}[Y^{\pi_0}|S_r^{\pi_0}, A_r^{\pi_0} = i] = \mathbf{E}[Y^{\pi_0}|A_{r+1}^{\pi_0} = i-1]$ holds since if at slot r there was a success and the number of active nodes was i, then at slot $r+1$ the number of active nodes must be $i-1$. Lastly, in the last step we used the equality $\mathbf{E}[Y^{\pi_0}|A_{r+1}^{\pi_0} = i-1] = \mathbf{E}[Y^{\pi_1}|A_{r+1}^{\pi_1} = i-1]$ that holds because both policies are the same in the interval from $r+1$ to t and we are conditioning on the number of active nodes at the beginning of the interval. An analogous argument shows that $\mathbf{E}[Y^{\pi_0}|F_r^{\pi_0}] = \mathbf{E}[Y^{\pi_1}|F_r^{\pi_1}]$ is also true. Using the above results we can write the difference $\mathbf{E}[W_t^{\pi_1}] - \mathbf{E}[W_t^{\pi_0}]$ as follows:

$$\mathbf{E}[W_t^{\pi_1}] - \mathbf{E}[W_t^{\pi_0}] = (1+\beta)(\mathbf{P}[S_r^{\pi_1}] - \mathbf{P}[S_r^{\pi_0}]),$$

where $\beta = \mathbf{E}[Y^{\pi_1}|S_r^{\pi_1}] - \mathbf{E}[Y^{\pi_1}|F_r^{\pi_1}] = \mathbf{E}[Y^{\pi_0}|S_r^{\pi_0}] - \mathbf{E}[Y^{\pi_0}|F_r^{\pi_0}]$. Because π_1 uses the greedy rule, we know that $\mathbf{P}[S_r^{\pi_1}] - \mathbf{P}[S_r^{\pi_0}] \geq 0$; therefore, to finish the proof, it only remains to show that β is greater or equal to -1:

$$\begin{aligned}-\beta &= \mathbf{E}[Y^{\pi_1}|F_r^{\pi_1}] - \mathbf{E}[Y^{\pi_1}|S_r^{\pi_1}] \\ &= \sum_{i=0}^{n} \mathbf{P}(A_r^{\pi_1} = i)(\mathbf{E}[Y^{\pi_1}|F_r^{\pi_1}, A_r^{\pi_1} = i] \\ &\quad - \mathbf{E}[Y^{\pi_1}|S_r^{\pi_1}, A_r^{\pi_1} = i]) \\ &= \sum_{i=0}^{n} \mathbf{P}(A_r^{\pi_1} = i)(\mathbf{E}[Y^{\pi_1}|A_{r+1}^{\pi_1} = i] \\ &\quad - \mathbf{E}[Y^{\pi_1}|A_{r+1}^{\pi_1} = i-1])\end{aligned}$$

We can now apply Proposition 2 (see Appendix A) to the difference inside the parenthesis since one can see $\mathbf{E}[Y^{\pi_1}|A_{r+1}^{\pi_1} = i]$ as the expected value of a new process that follows policy π_1 starting from slot $r+1$ and where the initial number of nodes is i. This gives $-\beta \leq 1$.

We can apply this exchange argument until transforming π_0 into the greedy policy, completing the proof. □

Corollary 1: If the optimization problem

$$g_t = argmax_{q\in[0,1]} \sum_{i=0}^{n} iq(1-q)^{i-1}\mathbf{a}[i]$$

is solved to optimality, then continuous GOAL-CR computes a sequence of access probabilities that minimize the expected value of the number of active nodes for all time slots t.

Proof: The proof of correctness of GOAL-CR only requires that the algorithm selects the optimal access probability at each slot t. In the case when B is finite, the optimality is easily guaranteed by iterating over all possible values in B. Therefore, if in the continuous case we assume that we have a method for computing the optimal access probability in the whole interval [0, 1], then the proof of correctness of GOAL-CR still follows unchanged. □

IV. ROBUSTNESS OF THE GREEDY ALGORITHM

As we saw in Section III, one of the disadvantages of the greedy algorithm is that its optimality is based on the supposition that all nodes start executing the policy at the same time. However, there might be scenarios where this supposition does not hold, like in the wildfire example discussed in Section I. We ran experiments where we simulated delays with a geometric distribution of parameter ϕ_d and observed that, although the resolution times obtained by policies generated using the greedy algorithm increase with ϕ_d, they still outperform the resolution times obtained when using a fixed optimal probability.

Another disadvantage of the greedy algorithm is that it requires the nodes to have enough memory to store the whole sequence of access probabilities, which might be infeasible for very low-memory sensors. In the next section, we address this difficulty by proposing a procedure to build a function of few parameters that approximates the policy obtained by GOAL-CR. With this approach, the nodes only need to store the current time slot and the parameters of the function to get the access probability p_t. Moreover, broadcasting a parameterized function consumes less resources than broadcasting a whole policy; hence, this solution is also of practical interest in scenarios where the sink node has to compute and share the access policy. Our simulation results show average contention resolution times close to the ones obtained with GOAL-CR using just two parameters.

V. LOW-MEMORY NODES

The last decades have seen the successful miniaturization of sensors for its use in WSNs. Unfortunately, small sensors usually have storage limitations because memory is the primary determinant of both chip area and energy budget [13]. Therefore, it is not practical for small sensor devices to store a whole sequence of access probabilities. Nevertheless, the policy computed by GOAL-CR can still be used to build a function

$$p(t|\alpha_1, \cdots, \alpha_k)$$

of few parameters $\alpha_1, \cdots, \alpha_k$ that gives the access probabilities for each time slot t. To approximate the sequence of access probabilities obtained by GOAL-CR, we considered four different models and used the standard weighted nonlinear least squares method [14] to optimize their parameters. More precisely, our procedure consists of choosing a parameterized model $p(t|\alpha_1, \alpha_2)$, a sequence of weights $\{w_t\}$, and then minimizing the weighted square error

$$\sum_{t=1}^{T} w_t(p(t|\alpha_1, \alpha_2) - p_t)^2.$$

TABLE 1. **Standard growth functions that we propose to fit the greedy policies. The hyperparameter *n* is the expected size of the network.**

1. Sigmoidal growth
$p(t\|\alpha_1,\alpha_2) = \frac{1}{n} + \left(1-\frac{1}{n}\right)\left(\frac{1}{1+exp(-\alpha_1(t-\frac{\alpha_2+3\cdot n}{2}))}\right)$
2. Power-law growth
$p(t\|\alpha_1,\alpha_2) = \alpha_1 \cdot t^{\alpha_2}$
3. Linear growth
$p(t\|\alpha_1,\alpha_2) = \max\left(\frac{1}{n}, \alpha_1 \cdot t + \alpha_2\right)$
4. Exponential growth
$p(t\|\alpha) = \frac{1}{n} \cdot \alpha^t$

We utilized the weighted version since we needed to give more importance to the access probabilities as they get closer to the beginning of the sequence. This is because the optimality of the access probability p_t computed by GOAL-CR at slot t is based on the assumption that the correct probabilities $\{p_1, \cdots, p_{t-1}\}$ where used from slot 1 to slot $t-1$, thus an error in the access probability at slot t would affect the optimality of all subsequent steps. For this reason, we selected weights that decay exponentially fast with t, i.e., $w_t = \gamma^t$ where γ is an hyperparameter in (0, 1] that measures the exponential decay rate.

We tested the parameterized models of the standard growth functions of Table 1 using the actual average contention resolution time obtained using simulations (see the Experimental Results section) as performance measure. Specifically, for each model we:

1) solved the weighted least squares problem for values of γ between (0, 1] in increments of 0.01,
2) evaluated the resulting functions with respect to the average contention resolution time, and

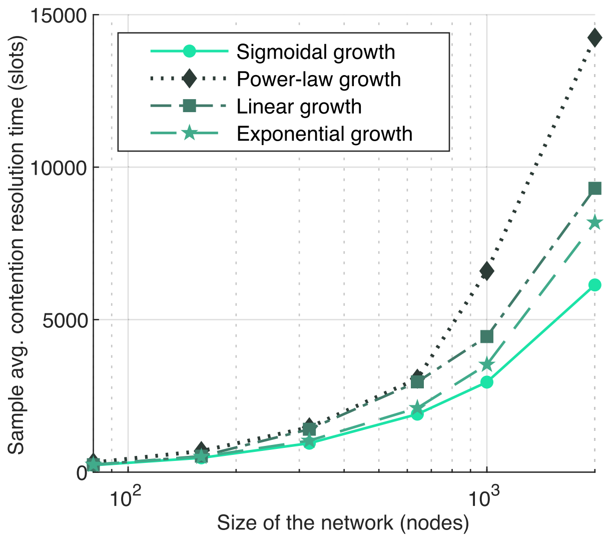

FIGURE 3. Sample avg. contention resolution time for networks of different sizes. The nodes use a parameterized function to adjust their access probability during the run-time of the contention.

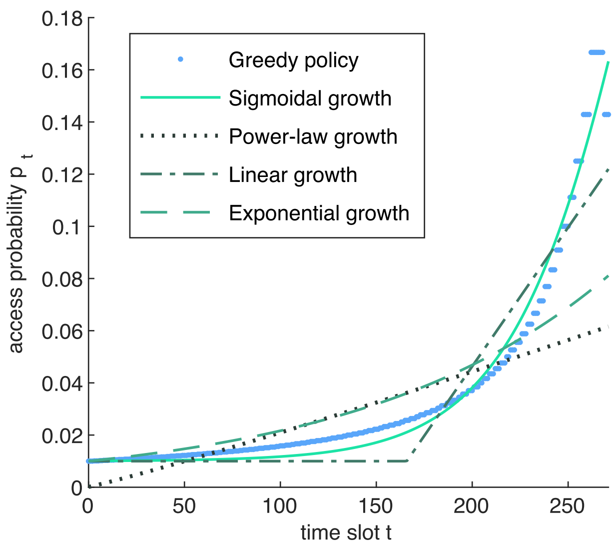

FIGURE 4. Parametrized functions that approximate a greedy policy computed to solve the contention in a network of size $n = 100$.

3) picked the values of the parameters obtained for the value of γ that gave the smallest average contention resolution time.

Fig. 3 shows the average contention resolution times of all the functions in Table 1, where the specific parameters were selected using the above procedure. Observe that the sigmoid function had the best performance for all values of n, which is consistent with Fig. 4 that shows the plots of all the functions and the optimal policy obtained by GOAL-CR for $n = 100$. Here, the sigmoid function is the one that best captures the trend of the greedy policy. In the next section we show, using simulations, that the performance of the sigmoid function (hereafter defined as growth rule) is still better than the performance of the protocol that uses a fixed optimal probability and slightly worse than the performance of GOAL-CR.

VI. EXPERIMENTAL RESULTS

We used simulations to evaluate the performance of GOAL-CR in terms of the average number of time slots needed for all the n active nodes to access the resource successfully, i.e., the average contention resolution time $D_{CR}(n)$. First, we built a platform that emulates random accesses to a shared resource; in our platform, time is divided into discrete slots, nodes attempt to access the resource at each time slot independently of each other using an access probability p_t, and the contention finishes when every node has accessed the resource successfully. Also, in order to focus on MAC performance, the channel is assumed to be ideal (i.e., multiple access interference is the only source of errors). Next, we allowed nodes of networks of different sizes to follow the protocols of Table 2 in scenarios of different nature. In our first scenario, we assumed that the number of active nodes at slot 0 is known with certainty and that the nodes start to compete for the resource at the same time. Then, we studied the performance of GOAL-CR in scenarios in which the nodes start to compete asynchronously. Finally, we examined the capability of GOAL-CR to handle uncertainty in the initial number active nodes.

The code for all experiments is written in MATLAB and can be found at https://github.com/ItzelOlivos/GOAL-CR.

A. THE INITIAL NUMBER OF NODES IS KNOWN

In our first testing scenario, we assumed that the size of the network is known with certainty. Accordingly, GOAL-CR receives a probability distribution concentrated on the total number of nodes as initial belief. We considered networks from 80 to 4000 nodes. Fig. 5 shows the sample average and the sample standard deviation, over 1000 independent simulations, of the contention resolution times obtained by each of the five protocols described in Table 2. Observe that even when the available information is just the probability distribution of active nodes at each time slot, continuous GOAL-CR achieves a $D_{CR}(n)$ close to $\Theta(n)$, which is asymptotically optimal; besides, observe how the performance of continuous GOAL-CR is closely followed by the basic version of GOAL-CR that computes policies using access probabilities that are restricted to be in the set $B = \{1/n, 1/(n-1), \cdots, 1/2, 1\}$. As for the case of the growth rule, it performs better than the fixed optimal probability and only slightly worse than the greedy policies. The figure also shows that the use of a fixed probability is adequate only when the size of the network is small because it reduces the complexity of the nodes and the network performance is similar to that of the schemes with variable probabilities.

B. RANDOM INITIAL DELAYS

The optimality of the greedy algorithm is based on the supposition that all nodes start executing the policy at the same time.

TABLE 2. Implementation description of the proposed protocols and state-of-the-art alternatives.

1. Perfect information
Nodes know with certainty the current number of active nodes A_t and attempt to access the resource with an optimal probability $p_t = 1/A_t$, achieving a $D_{CR}(n)$ in $\Theta(n)$. This assumption is hard to fulfill in real-world applications; however, it maximizes channel utilization and allows the derivation of a lower bound for the contention resolution problem.
2. GOAL-CR
Nodes adjust their access probability at each time slot following the policy π computed according to Algorithm 1. Since GOAL-CR computes sequences of access probabilities of length T, it might be the case that some nodes exhaust all values in π and still do not acquire the resource successfully; if so, these nodes use $\pi(T)$ until they succeed. To prevent those cases, we chose a sufficiently large value of T so that the expected number of active nodes at slot T is close to zero, i.e., $\sum_i i\mathbf{P}(A_T^\pi = i) \approx 0$.
3. Continuous GOAL-CR
Nodes adjust their access probability at each time slot following the policy π computed according to Algorithm 2. As in the discrete version of GOAL-CR, the nodes that remain active at slot T use $\pi(T)$ until they succeed. Likewise, we chose a sufficiently large value of T so that $\sum_i i\mathbf{P}(A_T^\pi = i) \approx 0$.
4. Growth rule
Nodes compete for the resource with an access probability p_t given by the the sigmoidal growth rule $$p_t = \frac{1}{n} + \left(1 - \frac{1}{n}\right)\left(\frac{1}{1 + exp(-\alpha_1(t - \frac{\alpha_2 + 3\cdot n}{2}))}\right)$$ The parameters α_1 and α_2 are chosen as described in section V.
5. Fixed optimal probability
Nodes try to access the resource with a fixed probability τ, computed at the beginning of the network operation, that minimizes the contention resolution time, i.e., the value of τ that minimizes $$D_{CR}(n) = \sum_{i=1}^{n} \frac{1}{i\tau(1-\tau)^{i-1}}.$$ We use the interior-point method provided by the Matlab Optimization Toolbox to compute the optimal value of τ. In Appendix B-2, we show that this protocol achieves a $D_{CR}(n)$ in $O(n \log n)$ and $\omega(n^{1-\epsilon} \log n)$ for any $\epsilon > 0$.

However, there might be scenarios where this supposition does not hold strictly and nodes start contending for the resource after a small delay. In order to study the performance of GOAL-CR against asynchronous starts, we ran experiments using a geometric distribution with parameter ϕ_d, i.e., each node starts to compete for the resource after i slots (where i is a random variable with mean ϕ_d^{-1} and variance $(1-\phi_d)/\phi_d^2$. We considered networks from 80 to 4000 nodes and measured the difference between the average contention resolution time of each protocol, $D_{CR}(n)$, and the lower bound of the problem $D_{CR}^*(n)$, i.e., the average contention resolution time achieved by the perfect-information protocol. As shown

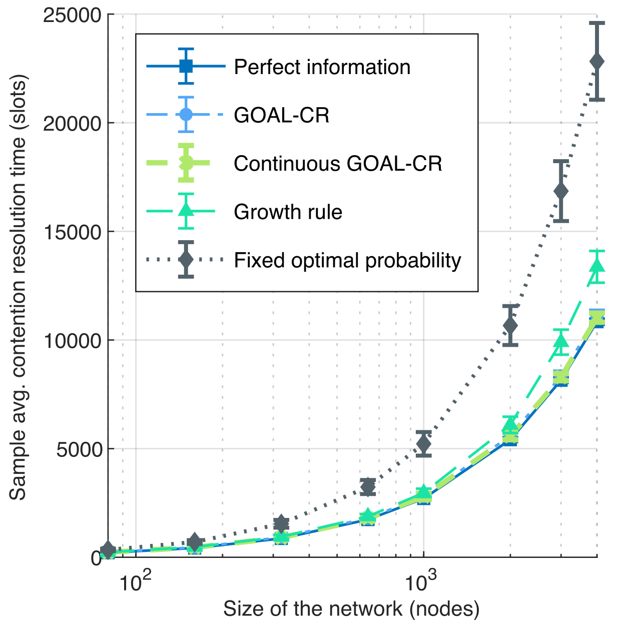

FIGURE 5. Sample avg. contention resolution time, obtained over 1000 independent simulations, for networks of different sizes in the scenario where the number of active nodes at time slot 0 is known with certainty.

in Fig. 6, the results obtained by the greedy policies, particularly for large-size networks, are still better than the ones obtained by the fixed-optimal-probability protocol even for delays with large expected values. Also, observe that the difference in performance of our proposals with respect to the perfect-information protocol becomes larger as ϕ_d^{-1} increases (i.e., as we increase both the magnitude and variance of the random delay), being the growth rule the most affected; however, both the basic GOAL-CR and its continuous version can handle significant amounts of noise.

C. UNCERTAINTY IN SIZE OF THE NETWORK

There are situations where the number of active nodes at time slot 0 is difficult to know with certainty; think, for instance, in a group of sensors that are deployed to track a wildfire and some of them are destroyed by flames. However, if the uncertainty can be described with a probability distribution, then GOAL-CR can still produce appropriate policies. In this section, we study the robustness of the protocols against uncertainty in the initial number of nodes. We ran experiments considering that nodes survive the deployment with probability λ_s. Thus, we adapted the protocols of Table 2 as follows: i) GOAL-CR and continuous GOAL-CR receive a binomial distribution with parameters n and λ_s as initial belief and ii) the growth rule and the fixed-optimal-probability protocol compute their respective access probabilities considering that the networks have $n\lambda_s$ nodes at slot 0. As in Section VI-B, we considered the difference $D_{CR}(n) - D_{CR}^*(n)$ as the measure of performance. Observe in Fig. 7 that the greedy policies maintain their difference with respect to the

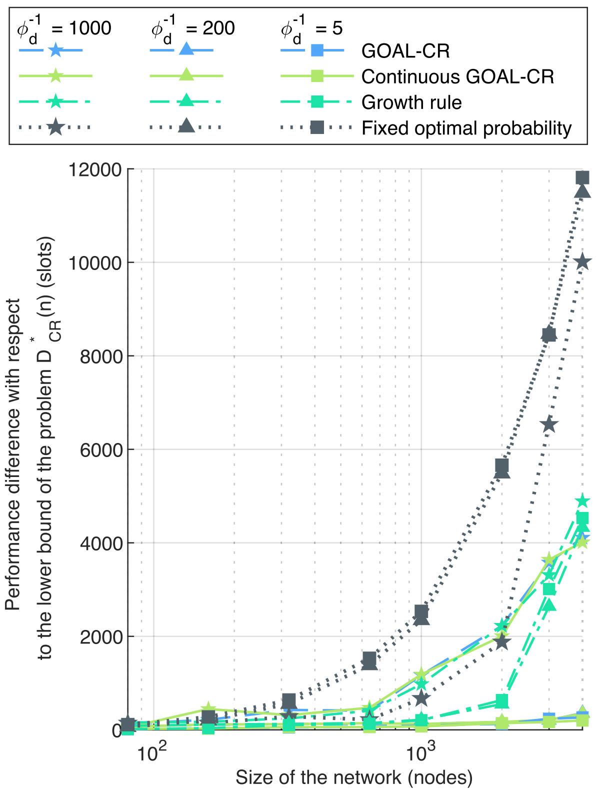

FIGURE 6. Performance of the protocols against asynchronous starts, over 1000 independent simulations, in terms of the difference $D_{CR}(n) - D^*_{CR}(n)$. The random delays follow a geometric distribution with parameter ϕ_d, i.e., each node starts to compete for the resource after ϕ_d^{-1} slots in expectation.

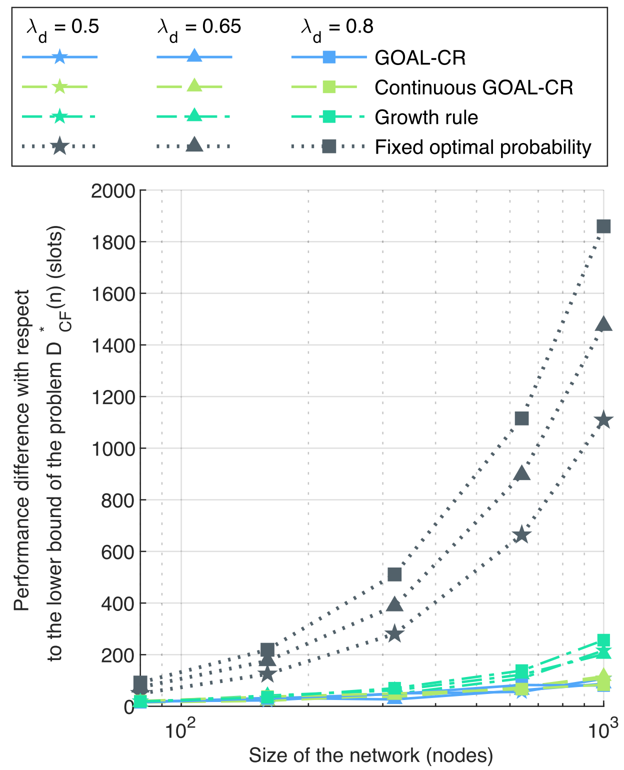

FIGURE 7. Performance of the protocols against randomness in the size of the network, over 1000 independent simulations, in terms of the difference $D_{CR}(n) - D^*_{CR}(n)$. The initial belief of active nodes is a binomial distribution with parameters n and λ_s.

lower bound of the problem despite the increments in the variance of the initial number of active nodes. In contrast, the performance of the growth rule is affected as the variance increases; we presume that GOAL-CR produces more aggressive policies for small values of λ_s and that the growth rule cannot well approximate this behavior due to the simplicity of its form. It is interesting that the fixed policy improves its results as the randomness increases. We believe this behavior might be related with the symmetry of the initial distribution since the fixed optimal value is based only on the mean. To test this hypothesis, we also ran experiments where we controlled the asymmetry of the initial distribution of nodes.

We studied the performance of the protocols against asymmetric initial distributions as follows: each distribution is a discretized mixture of Gaussians with constant variance and proportion but where the distance between their means μ_1 and μ_2 vary (see Fig. 8(a)). GOAL-CR and continuous GOAL-CR received the discretized mixture of Gaussians as initial belief, whereas the growth rule and the fixed-optimal-probability protocol computed their respective access probabilities considering that the networks had $(\mu_1 + \mu_2)/2$ nodes at slot 0. Fig. 8(b) shows how the protocol that uses a fixed access probability can handle a certain degree of asymmetry but then loses dramatically against the greedy policies and the growth rule.

VII. CONCLUSIONS

The proposed algorithm solves a variant of the standard contention resolution problem in which the nodes need to access the shared resource only once and where the objective is to minimize the time it takes for all the nodes to successfully access the resource. We model the problem as a POMDP, and our formal analysis shows that the resulting specific instances can be optimally solved in polynomial time using a greedy algorithm. GOAL-CR can compute access policies that minimize the expected contention resolution time, $D_{CR}(n)$, without requiring to know the precise number of active nodes at run-time. This is particularly important in the context of WSNs because it greatly reduces the complexity of the nodes. We also showed that the optimal value of $D_{CR}(n)$ is in $\Theta(n)$ when the number of active nodes is known with certainty and $O(n \log n)$ and $\omega(n^{1-\epsilon} \log n)$ for any $\epsilon > 0$ when nodes attempt to access the resource with a fixed optimal probability. Additionally, we showed how to use GOAL-CR in scenarios where nodes do not have enough memory to store the complete sequence of access probabilities. Lastly, our experimental results show that the performance of the proposed algorithm is equivalent to that of the algorithm with perfect information about the current number of active nodes when there are not additional sources of randomness, and still better than the alternatives when both there is uncertainty in the initial number of active nodes and when the nodes

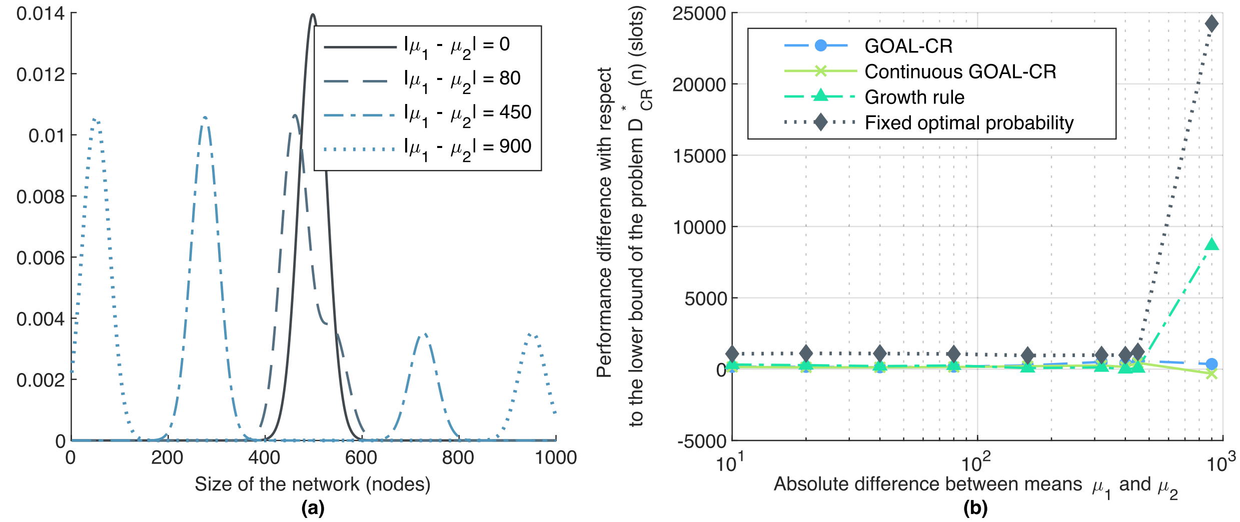

FIGURE 8. (a) Mixture of Gaussians describing the initial number of active nodes. (b) Performance of the protocols against randomness in the size of the network, over 1000 independent simulations, in terms of the difference $D_{CR}(n) - D^*_{CR}(n)$. The initial belief of active nodes is a discretized mixture of Gaussians with constant variance and proportion but where the distance between their means μ_1 and μ_2 vary.

start the contention after a small random delay. In future works, we intend to formally show the relation between the perfect-information protocol and the policies obtained by GOAL-CR. In specific, we want to show that the expected contention resolution time of GOAL-CR is $O(n)$. Additionally, we want to exploit the capabilities of reinforcement learning methods to supplement our approach by automatically discovering the underlying processes that explain more dynamic scenarios, e.g., nodes transmitting safety-related warning messages in Vehicular Ad-Hoc Networks.

APPENDIX A
FOR THE CORRECTNESS OF GOAL-CR

Proposition 1: Given the belief of active nodes at slot t, the problem of minimizing the expected number of active nodes at slot t + 1 is equivalent to the problem of maximizing the probability of having a successful access at slot t.

Proof: Let $\mathbf{E}[A^\pi_{t+1}] = \sum_i i\mathbf{P}(A^\pi_{t+1} = i)$ be the expected number of active nodes at slot $t+1$ when using the policy π. Since $\mathbf{P}(A^\pi_{t+1} = i) = \mathbf{P}(A^\pi_{t+1} = i | A^\pi_t = i)\mathbf{P}(A^\pi_t = i) + \mathbf{P}(A^\pi_{t+1} = i | A^\pi_t = +1\mathbf{P}(A^\pi_t = i+1)$, $\mathbf{E}[A^\pi_{t+1}]$ can be expressed as:

$$\mathbf{E}[A^\pi_{t+1}] = \sum_{i=0}^{n} i(1 - ip_t(1-p_t)^{i-1})\,\mathbf{P}(A^\pi_t = i) + \sum_{i=0}^{n} i(i+1)\,p_t(1-p_t)^i\mathbf{P}\,(A^\pi_t = i+1) \quad (2)$$

Rearranging the indexes of the second term of (2), and noticing that $\mathbf{P}(A^\pi_t = n+1) = 0$ since the value of A^π_t is at most n, yields:

$$\mathbf{E}[A^\pi_{t+1}] = \sum_{i=0}^{n} i\mathbf{P}(A^\pi_t = i) - \sum_{i=0}^{n} ip_t(1-p_t)^{i-1}\mathbf{P}(A^\pi_t = i) \quad (3)$$

The first term in (3) corresponds to the expected value of the number of active nodes at slot t and the second term is equal to minus the probability of a successful access given the probability distribution of active nodes at slot t. The probability of a successful access depends only on p_t; thus, to minimize $\mathbf{E}[A^\pi_{t+1}]$ we just need to maximize the probability of a successful access at slot t. □

Proposition 2: For any policy π

$$\mathbf{E}[W^\pi_t | A^\pi_0 = i] - \mathbf{E}[W^\pi_t | A^\pi_0 = i-1] \leq 1 \quad (4)$$

Proof: The proof is by induction on t. When $t = 0$, $\mathbf{E}[W^\pi_0 | A^\pi_0 = i]$ is equal to the probability of success at slot 0 given i active nodes, i.e., $\mathbf{P}[S^\pi_0 | A^\pi_0 = i] = ip_0(1-p_0)^{i-1}$, where S^π_0 is an indicator r. v. that takes the value of 1 if there is a successful access at slot 0. Similarly $\mathbf{E}[W^\pi_0 | A^\pi_0 = i-1] = \mathbf{P}[S^\pi_0 | A^\pi_0 = i-1] = (i-1)p_0(1-p_0)^{i-2}$. Since both are probabilities, its difference is less than 1.

For the induction step, we will use the following notation: let $\pi(r)$ denote the policy that is equal to policy π but without the first $r-1$ probabilities, i.e., the l-th probability of access in policy $\pi(r)$ is equal to the $(r+l)$-th probability in policy π. We expand the first expected value in the difference of (4) using the law of total expectation over the indicator r. v. S^π_0:

$$\begin{aligned}
&\mathbf{E}[W^\pi_t | A^\pi_0 = i] - \mathbf{E}[W^\pi_t | A^\pi_0 = i-1] \\
&\quad = \mathbf{P}[S^\pi_0 | A^\pi_0 = i](1 + \mathbf{E}[W^{\pi(1)}_{t-1} | A^{\pi(1)}_0 = i-1]) \\
&\qquad + (1 - \mathbf{P}[S^\pi_0 | A^\pi_0 = i])\mathbf{E}[W^{\pi(1)}_{t-1} | A^{\pi(1)}_0 = i] \\
&\qquad - \mathbf{E}[W^\pi_t | A^\pi_0 = i-1].
\end{aligned}$$

Observe that $\mathbf{E}[W^\pi_t | A^\pi_0 = i-1] \geq \mathbf{E}[W^{\pi(1)}_{t-1} | A^{\pi(1)}_0 = i-1]$. Therefore,

$$\begin{aligned}
&\mathbf{E}[W^\pi_t | A^\pi_0 = i] - \mathbf{E}[W^\pi_t | A^\pi_0 \\
&\quad = i-1] \leq \mathbf{P}[S^\pi_0 | A^\pi_0 = i] \\
&\qquad + (1 - \mathbf{P}[S^\pi_0 | A^\pi_0 = i])(\mathbf{E}[W^{\pi(1)}_{t-1} | A^{\pi(1)}_0 = i]
\end{aligned}$$

$$- \mathbf{E}[W_{t-1}^{\pi(1)} | A_0^{\pi(1)} = i-1]) \leq \mathbf{P}[S_0^{\pi} | A_0^{\pi} = i] + (1 - \mathbf{P}[S_0^{\pi} | A_0^{\pi} = i]) = 1,$$

where in the last step we used the induction hypothesis. □

APPENDIX B
ASYMPTOTICS FOR THE COMPLETE INFORMATION AND FIXED OPTIMAL PROBABILITY PROTOCOLS

Lemma 1: The optimal value of $D_{CR}(n)$ is in $\Theta(n)$ when the current number of active nodes A_t is known.

Proof: Let X_k be the r. v. that counts the number of time slots between the $(i-1)^{th}$ and the i^{th} successful transmission; hence, $D_{CR}(n) = \sum_{k=1}^{n} \mathbf{E}[X_k]$. The optimal value of $D_{CR}(n)$ is achieved by maximizing the probability of success $p = A_t\tau(1-\tau)^{A_t-1}$ of each X_k, which results in $\tau = 1/A_t$. Then

$$D_{CR}(n) = \sum_{A_t=1}^{n} (1 - 1/A_t)^{1-A_t}.$$

The function $(1-1/n)^{n-1}$ decreases monotonically from $1/2$ down to $1/e$ as n increases from 2. Thus, the reciprocal of the function satisfies $(1-1/n)^{1-n} \leq e$ which results in

$$\sum_{A_t=1}^{n} (1 - 1/A_t)^{1-A_t} \leq \sum_{A_t=1}^{n} e = en.$$

Hence, $D_{CR}(n)$ is $O(n)$. To show that $D_{CR}(n)$ is $\Omega(n)$ notice that $(1-1/A_t)^{1-A_t} \geq 1/2$, i.e. $\sum_{A_t=1}^{n}(1-1/A_t)^{1-A_t} \geq 2n$. □

Lemma 2: The optimal value of $D_{CR}(n)$ is $O(n \log n)$ and $\omega(n^{1-\epsilon} \log n)$ for any $\epsilon > 0$ when the nodes attempt to access the resource with a fixed optimal probability τ.

Proof: The expected value of the contention resolution time $D_{CR}(n)$ is given by $D_{CR}(n) = \sum_{i=1}^{n}(i\tau(1-\tau)^{i-1})^{-1}$. Let $\tau = c/n$ and let c^* be the value of c that minimizes $D_{CR}(n)$ i.e.,

$$\frac{n}{c^*} \sum_{i=1}^{n} \frac{1}{i(1-c^*/n)^{i-1}} \leq \frac{n}{c} \sum_{i=1}^{n} \frac{1}{i(1-c/n)^{i-1}}$$

for all $c \in (0, n]$. In particular when $c = 1$ we get

$$\frac{n}{c^*} \sum_{i=1}^{n} \frac{1}{i(1-c^*/n)^{i-1}} \leq n \sum_{i=1}^{n} \frac{1}{i(1-1/n)^{i-1}}.$$

Using $(1-1/n)^{1-m} \leq (1-1/n)^{1-n} \leq e$ for $m \leq n$ results in

$$n \sum_{i=1}^{n} \frac{1}{i(1-1/n)^{i-1}} \leq en \sum_{i=1}^{n} \frac{1}{i} \leq en\,(\log n + 1)$$

where in the last inequality we used the fact that the harmonic sum satisfies $\log(n+1) \leq \sum_{i=1}^{n} 1/i \leq 1 + \log(n)$. Therefore, $D_{CR}(n)$ is $O(n \log n)$.

To show that $D_{CR}(n)$ is $\omega(n^{1-\epsilon} \log n)$ for any $\epsilon > 0$, we compute two lower bounds that depend on c. The first one focuses on the last term in the summation expressing the expected value of the contention resolution: $c^{-1}(1-c/n)^{1-i} \geq c^{-1}e^{c-c/n}$. The second one uses the inequality $(1-c/n)^{1-i} \leq 1$ to show that

$$\frac{n}{c} \sum_{i=1}^{n} \frac{1}{i(1-c/n)^{i-1}} \geq \frac{n}{c} \sum_{i=1}^{n} \frac{1}{i} \geq (n/c) \log n.$$

The first lower bound $c^{-1}e^{c-c/n}$ increases exponentially with c whereas the second one $(n/c)\log n$ decreases inversely with c. This means that the optimal value of c^* must be in $o(n^{\epsilon})$ otherwise the expected contention resolution time would grow exponentially contradicting the upper bound $O(n \log n)$ computed before. Hence the optimal expected contention resolution time must be, using the second lower bound, $\omega(n^{1-\epsilon} \log n)$ for any $\epsilon > 0$. □

REFERENCES

[1] J. Kleinberg and E. Tardos, *Algorithm Design*. Boston, MA, USA: Addison-Wesley Longman Publishing, 2005.

[2] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[3] S. K. Singh, P. Kumar, and J. P. Singh, "A survey on successors of LEACH protocol," *IEEE Access*, vol. 5, pp. 4298–4328, 2017.

[4] M. E. Rivero-Angeles and G. Rubino, "Transmission probability strategies for cluster-based event-driven wireless sensor networks," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Oct. 2017, pp. 416–419.

[5] Y. H. Bae, "Analysis of optimal random access for broadcasting with deadline in cognitive radio networks," *IEEE Commun. Lett.*, vol. 17, no. 3, pp. 573–575, Mar. 2013.

[6] R. Bruno, M. Conti, and E. Gregori, "A simple protocol for the dynamic tuning of the backoff mechanism in IEEE 802.11 networks," *Comput. Netw.*, vol. 37, no. 1, pp. 33–44, 2001.

[7] M. Albalt and Q. Nasir, "Adaptive backoff algorithm for IEEE 802.11 MAC protocol," *Int. J. Commun., Netw. Syst. Sci.*, vol. 2, no. 4, p. 300, 2009.

[8] R. Bruno, M. Conti, and E. Gregori, "Optimal capacity of p-persistent CSMA protocols," *IEEE Commun. Lett.*, vol. 7, no. 3, pp. 139–141, Mar. 2003.

[9] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998, vol. 135.

[10] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Auton. Agents Multi-Agent Syst.*, vol. 27, no. 1, pp. 1–51, Jul. 2012.

[11] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of Markov decision processes," *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, 1987.

[12] A. Kobel, F. Rouillier, and M. Sagraloff, *Computing Real Roots of Real Polynomials... and Now for Real!* New York, NY, USA: ACM, 2016, pp. 303–310.

[13] L. Benini, E. Farella, and C. Guiducci, "Wireless sensor networks: Enabling technology for ambient intelligence," *Microelectron. J.*, vol. 37, no. 12, pp. 1639–1649, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026269206001728

[14] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

ITZEL C. OLIVOS-CASTILLO received the B.Sc. degree in telematics engineering and the M.Sc. degree in computer science from the Instituto Politécnico Nacional, Mexico City, Mexico, in 2015 and 2017, respectively. She is currently a Fulbright-García Robles Grantee. Her research interests include the intersection of algorithms' design, reinforcement learning, and computational cognitive science.

RICARDO MENCHACA-MENDEZ received the B.S. degree in computer science from the Escuela Superior de Cómputo, Mexico City, Mexico, in 2001, the M.S. degree from the Instituto Politécnico Nacional, Mexico City, in 2005, and the Ph.D. degree in computer science from the University of California at Santa Cruz, in 2013. He is currently a Professor with the Centro de Investigación en Computación, Instituto Politécnico Nacional.

ROLANDO MENCHACA-MENDEZ received the B.S. degree in electronic engineering from the Universidad Autónoma Metropolitana, Mexico City, Mexico, in 1997, the M.S. degree from the Instituto Politécnico Nacional, Mexico City, in 1999, and the Ph.D. degree in computer engineering from the University of California at Santa Cruz, in 2009. He is currently a Professor and the Head of the Network and Data Science Laboratory, Centro de Investigación en Computación, Instituto Politécnico Nacional.

MARCELO M. CARVALHO (M'02) received the B.Sc. degree in electrical engineering from the Federal University of Pernambuco, Brazil, in 1995, the M.Sc. degree in electrical and computer engineering from the State University of Campinas, Brazil, in 1998, the M.Sc. degree in electrical and computer engineering from the University of California at Santa Barbara, USA, in 2003, and the Ph.D. degree in computer engineering from the University of California at Santa Cruz, USA, in 2006. He is currently an Assistant Professor with the Department of Electrical Engineering, University of Brasília, Brazil. His research interests are in the fields of wireless networking, ad hoc and sensor networks, and multimedia communications. He is a member of the ACM.

MARIO E. RIVERO-ANGELES (S'00–M'04) was born in Mexico City, Mexico, in 1976. He received the B.Sc. degree from the Universidad Autónoma Metropolitana, Mexico, in 1998, and the M.Sc. and Ph.D. degrees in electrical engineering form CINVESTAV-IPN, in 2000 and 2006, respectively. He was a Postdoctoral Fellow with the Dyonisos Research Project Institut National de Recherche en Informatique et en Automatique, Rennes, France, from 2007 to 2010. He is currently a Professor with the Instituto Politécnico Nacional, where he has been with the Centro de Investigación en Computación, Mexico, since 2002. His research interests include random access protocols and data transmission in cellular networks, P2P networks, and wireless sensor networks.