

Local control of cognitive radio networks

Christian Doerr · Dirk Grunwald · Douglas C. Sicker

Received: 23 June 2008 / Accepted: 26 March 2009 / Published online: 16 May 2009
© The Author(s) 2009. This article is published with open access at Springerlink.com

Abstract In a network deployment, a cognitive radio will have to perform two fundamental tasks. First, each cognitive radio needs to optimize its internal operation, and second, it needs to derive a configuration that will enable and optimize communication with other nodes in the network. This latter requirement, however, relies on knowledge about the other nodes' current configuration settings, which needs to be incorporated into this decision-making process. Collecting and distributing such global knowledge is, however, a difficult and costly process, which, in the past, has been approached by introducing a centralized control authority, distributed negotiation policies, or a dedicated coordination channel in the network, each resulting in vulnerability and scaling issues. In this paper, we propose an alternative approach to the global configuration of a cognitive radio network that eliminates the need to collect global network state information and, instead, uses local information for its decision making process. This technique is built upon the principles of swarm

intelligence, as seen in schools of fish and flocks of birds, and allows for efficient and robust coordination of a cognitive radio network in a variety of tasks. We have implemented a working prototype showing the feasibility of this technique in two simulation environments and in a hardware testbed, and find that a solution based on swarm intelligence is well suited to interoperate in heterogeneous deployment environments with other control algorithms, requires low computational overhead, and scales with the number of nodes and the amount of spectrum, thus making it a versatile control algorithm for many deployment scenarios.

Keywords Cognitive radio networks · Swarm intelligence · Emergent behavior · Local independent control · Dynamic spectrum access

1 Introduction

Emerging cognitive radio networks will be faced with a broad set of sensing, learning, and adaptation requirements, and each of these requirements becomes ever more complex as the number of network nodes and network diversity increases. Beyond optimizing its internal operation and reconfiguration procedures, it is necessary for a cognitive radio to incorporate information about neighboring wireless network nodes and other sources of interference. As the number of nodes increases, it is difficult and costly to obtain sufficient information for making appropriate frequency usage decisions, particularly based on approaches that share information globally or that require centralized control. Due to the difficulties in obtaining global knowledge about spectrum usage across a radio network, it is

C. Doerr (✉) · D. Grunwald · D. C. Sicker
Department of Computer Science,
University of Colorado at Boulder,
430 UCB, Boulder, CO 80309, USA
e-mail: Christian.Doerr@colorado.edu

D. Grunwald
e-mail: Dirk.Grunwald@colorado.edu

D. C. Sicker
e-mail: Douglas.Sicker@colorado.edu

Present Address:
C. Doerr
Department of Telecommunication, TU Delft,
Delft, The Netherlands
e-mail: c.doerr@tudelft.nl

worthwhile to explore whether adequate configuration decisions can also be determined when there is no central authority to determine spectrum usage or when the network nodes do not possess knowledge of the entire network topology. Here, we propose a method that relies on information that can be locally observed by each node. If it is possible to obtain sufficient network coordination by basing the decision on local knowledge only, such a cognitive radio system would enjoy higher efficiency, better scalability, and increased robustness, as the bottleneck of explicit coordination communication is removed.

In this paper, we describe such a local control model based on the principles of swarm intelligence. The collective and seemingly complex behavior of swarming as expressed by schools of fish and flocks of birds is actually the result of a set of simple rules that are followed by each individual in the group. Yet, when each member observes these rules, everyone's local behavior and the resulting interactions between individuals give rise to a complex global behavior exhibiting properties beyond any behavioral response encoded in the original rule set. In this paper, we distill and translate the essential aspects of swarm intelligence to the problem of dynamic spectrum allocation. We show that a rather straightforward set of requirements can be adapted for swarm behavior to provide proper configuration decisions to small and large sets of cognitive radio networks operating in a variety of situations.

To examine the applicability of local control to cognitive radio networking, we first examine the requirements and challenges of dynamic spectrum usage. We then consider the problem of distributed cognitive radio coordination based on the principles of swarm intelligence. Next, we demonstrate how the principles of repulsion and attraction (concepts central to swarm intelligence) can be applied to cognitive radio coordination by presenting the results of a simulation based on these principles. To confirm our simulation results, we developed a working prototype of a cognitive radio network controlled by the principles of swarm behavior and demonstrated the feasibility of local control in a hardware testbed using off-the-shelf (commodity) network nodes. Finally, we investigate the algorithm's scalability in terms of the overall amount of accessible spectrum and determine to what extent the algorithm based on swarm intelligence may be assisted by simple additional scanner components, such as the low-cost WiSpy energy detector. In conclusion, we find that this type of control algorithm is well suited to efficiently configure wireless networks based on local information only and may be applied in a variety of deployment scenarios. The material presented in this paper is a

compilation and extension of previous results discussed in [9–11, 13], supplemented with additional details and new results.

1.1 The requirements and challenges of dynamic spectrum usage

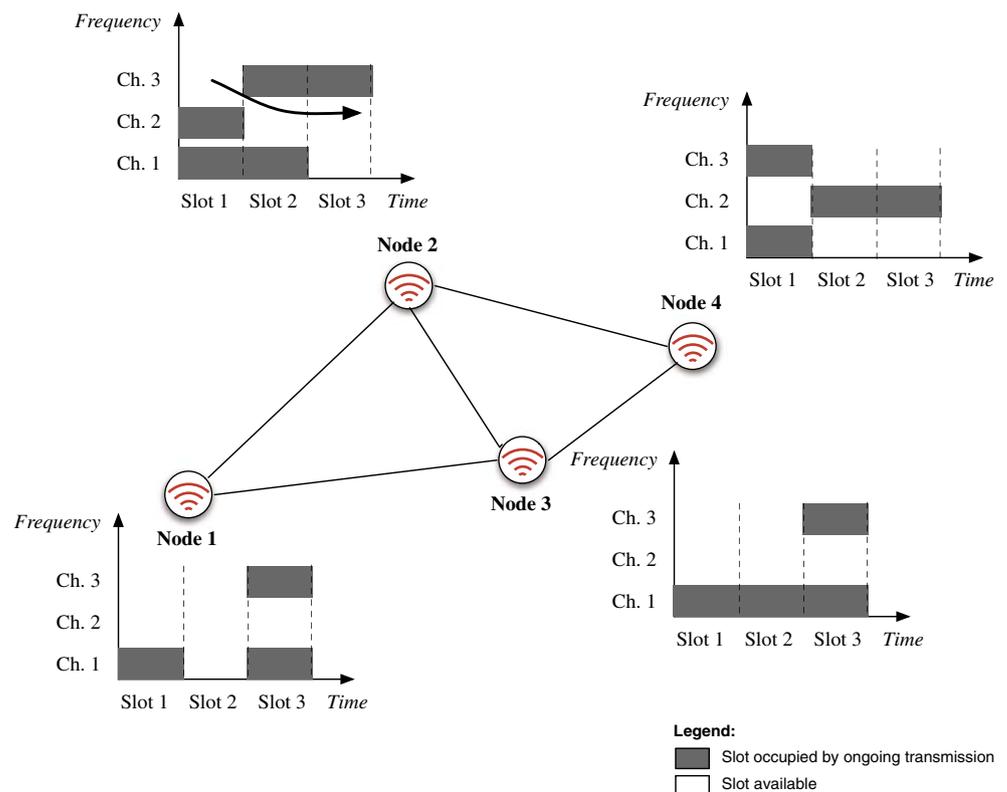
To start our discussion, we begin with an analysis of the requirements and challenges of dynamic spectrum access (DSA) as visualized in Fig. 1, which contains a small network of four cognitive radio nodes. In this example network, nodes 1, 2, and 3 are all in range of each other, while node 4 only has links to nodes 2 and 3.

Within the paradigm of DSA, a node is not assigned to a fixed frequency but, rather, scans the spectrum at run-time to determine and choose from the locally available frequencies. For the discussion in this simple example, we assume that each radio can only access three orthogonal channels and that spectrum availability changes in discrete time steps. In practice, a cognitive radio could have a significantly larger number of theoretically accessible channels, spectrum and its availability will change more fluid, and radios in the network will not make synchronized decisions but in their decision-making process rely on their own, local clock.

As can be seen in the figure, spectrum availability in the network is highly heterogeneous in that each node has a different set of available frequencies at any given time. While this example is limited in size and scope, it does help to formulate the requirements that a control algorithm for dynamic spectrum coordination will need to fulfill.

1. *React to spectrum changes* First, a control algorithm must be able to sense, identify, and reconfigure to use the locally available channels. As the spectrum availability changes, the algorithm must learn and update its allocation decisions to always reflect the current environmental conditions.
2. *Propagate important spectrum changes* Second, a radio cannot exist in isolation, but must be a part of a larger network structure. Therefore, it must take into account not only availability, but connectivity to other nodes. The frequency allocation should, therefore, be chosen in such a way that a link to a neighbor is always maintained if possible. Consider the example of Fig. 1. Here, node 1 has, during the first time slot, multiple options (frequency 2 or 3), but only channel 3 would allow it to connect to node 2 (and node 3). As time progresses, local frequency allocations may need to change even

Fig. 1 Distributed channel coordination of a cognitive radio network requires knowledge about the spectrum availability of each network node at any point in time



if local availability has remained constant. During time slot 2, frequency 3 becomes unavailable to node 2, thus requiring it to change to an alternative channel, frequency 2. Even though both nodes 1 and 3 may still continue to use the old frequency, it is beneficial for them to change as well, as the alternative offers better connectivity between the network nodes. Therefore, a cognitive radio control algorithm should be able to also learn status information from remote nodes and update its local decisions accordingly, if necessary.

3. *Coordination across multiple hops* Third, links and their frequency allocations should be coordinated across multiple hops to ensure routing connectivity among a set of geographically dispersed nodes. In traditional wireless networks operating on a fixed frequency, all communication may be either delivered directly or on routes found, for example, by a shortest path algorithm. In DSA networks, link availability changes with spectrum availability and a link may become temporarily unusable even though it is theoretically feasible and was previously accessible due to spectrum heterogeneity and spectrum dynamics.

Figure 1 visualizes this issue. Although always connected by a link to nodes 2 and 3, node 4 may only

use one of its theoretically feasible links as spectrum availability fluctuates, consequently requiring all nodes in the example to update their decisions when transmitting to node 4. Therefore, for proper data delivery, the nodes in the network must incorporate information about link availability into their decision-making process (either spectrum selection or routing), even if they are multiple hops away.

1.2 Spectrum-aware channel selection—a global coordination problem

In order to achieve an efficient channel allocation taking into account local spectrum availability, remote changes, and multiple-hop coordination, every node in the network would ideally have global knowledge about the network topology, link states, and spectrum availability of every other node. Such global knowledge could be achieved in two ways:

- (a) By introducing a central control authority that collects spectrum availability from all other nodes and distributes frequency assignments
- (b) By each node independently obtaining state information from all other nodes in the network and locally forming a decision

This process, however, would be difficult to implement in practice due to its cyclic nature: In order to collect state information from all nodes or between all nodes, a channel selection is needed that would enable all nodes to connect and communicate with each other. This allocation is, however, the result of the frequency selection process for which the network state information is being collected.

To overcome this issue, one could rely on a dedicated control channel on which channel-state and configuration information is being exchanged. There exist a number of ways to achieve this, for example, by setting aside a fixed (potentially licensed) channel, dynamically finding a frequency at run-time, or negotiating a dynamic coordination frequency using an out-of-band coordination channel such as the proposed 50-kHz E^2R CPC [1], each having certain drawbacks. A fixed frequency would need to be known beforehand by each network node and be available at all times; otherwise, coordination and continuous adaptation cannot be achieved. Such a fixed channel, however, could make the system vulnerable to deliberate or accidental jamming or interference, rendering the network inoperable, and in the case of licensed spectrum creating a significant cost. On the other hand, dynamically assigning a coordination channel would require a common algorithm that is employed by all network nodes for locating such a channel, and this may become difficult in a diverse and large network system, due to the heterogeneity of the nodes and spectrum heterogeneity so that such a common channel may not exist. It is possible to mix these approaches and create hybrid solutions, for example using a dedicated calling frequency to aid in the selection of a dynamic channel.

Due to these difficulties in collecting global knowledge, it may be worth exploring whether it is sufficient to base a node's decision-making process only on information that may be obtained locally. If it is possible to obtain adequate coordination using only local knowledge, we would be able to avoid the bottleneck of explicit centralized coordination and expensive data dissemination.

2 Related work

A number of groups have proposed a variety of control algorithms for managing and configuring cognitive radio networks. In the following, we summarize the current state of the art of cognitive radio management using the categorizations of central, distributed, and local control.

2.1 Centralized control

A number of centralized strategies build on the idea that spectrum allocation and node configuration can be best designed through exploiting global knowledge; therefore, these approaches rely on a centralized server that collects and exchanges information with clients. Raman et al. [23] determine the optimal link configuration using a linear programming approach. In this paper, a spectrum server calculates link schedules to maximize total throughput on the network. To prevent link starvation, the program considers minimum link rate constraints. Ileri et al. [15] developed a spectrum policy server that brokers frequency assignments. This system collects spectrum requests, matches them to spectrum availabilities, and makes use of demand-responsive pricing to ensure fairness in the frequency distribution while still maximizing profits for the spectrum holder.

2.2 Distributed control

To avoid the potential bottleneck and dependencies of a centralized server, several researchers have proposed distributed schemes. Cao and Zheng [5], for example, suggest to allocate spectrum distributively through local, incremental bargaining. As global optimization of spectrum allocations takes a considerable amount of time and resources, the authors react incrementally to changes by negotiating spectrum allocation conflicts on a one-to-one or one-to-many basis in local bargaining operations. This will speed up a conflict-free allocation of spectrum as only changes due to the recent mobility of users are remediated; on the other hand, the approach requires that, formerly, an initial robust solution had existed that was disturbed through user mobility and cannot be used to derive a system-wide resource allocation.

Neel and Reed [21] formulate a distributed channel assignment algorithm based on game theory. When this algorithm is then rolled out to a cognitive radio network, each node has implicit knowledge of a network-wide utility function, i.e., every node in the network associates the same benefit to the available spectrum. The nodes then individually make decisions on how to maximize their individual benefit depending on the decisions their surrounding neighbors have previously taken. If the utility function has been properly crafted and there exists an equilibrium point, the network will converge to a stable solution. We characterize Neel and Reed as a distributed approach because the standard game theoretic framework makes two fundamental assumptions on

the network nodes to achieve convergence: (1) all nodes need to have the same utility function and (2) the decision-making process is sequential. These assumptions therefore practically require a distributed protocol, even though the decisions could be made and observed locally, thus making the control algorithm in our scheme a distributed system.

Weingart et al. [27] use the design of experiments approach to find the best possible link configuration for a cognitive radio. In this work, a cognitive radio initiates a sequence of test transmissions to surrounding nodes, intending to find possible correlations between system parameters and their changes to network performance. The results can then be used to create a linear model identifying the best possible link configurations.

Another common approach to configure cognitive radios is the use of genetic algorithms as presented in Rondeau et al. [25]. Using this methodology, a utility function is explored through multipoint hill climbing to find the optimal configuration setting. Rondeau et al. [25] cover the details of this technique.

2.3 Local control

Beyond these centralized and distributed control schemes, there also exists what can be referred to as a localized independent control scheme. We characterize a local cognitive radio control algorithm as one where no explicit communication occurs with neighboring nodes, rather decisions are based on passive observations of local information. Aside from our prior work, we have found another cognitive control algorithm based on local control. In work by Zheng and Cao [28], the authors describe a self-management approach to a single radio node configuration based on local control; in this work, however, the coordination algorithm is not concerned with coordinating configuration decisions across the network or creating routes that will connect several nodes with each other.

3 Local coordination through the principles of swarm intelligence

The problem of distributed coordination that needs to be solved in a cognitive radio network is not too different from many coordination and cooperation tasks found in nature. In fact, there exist many instances in animal behavior where each individual of a larger group possesses only a piece of knowledge about the state of the system or the general environment, but where all pieces are needed to obtain a good solution. These

issues of distributed sensing, information aggregation, and decision making can be found across many species:

- In a group of schooling fish, each individual member of the swarm possesses information about food resources, obstacles, and dangers in its direct neighborhood, the area that the fish can monitor using its own senses. As there exist threats that are only detected by some individuals but yet form a danger to others, if not all members in the group, such information needs to be efficiently shared and spread among the group members so that the entire group may react to information that is only detected by a few.
- In cases where resources may need to be located in an area too large for a single individual to roam alone (for example, through chemotaxis), individuals in many species, such as, for example, Antarctic krill, team up and form large structures to efficiently detect and follow along resource gradients due to their enhanced, combined search capabilities [14, 22].
- When remote food sources have been found by scouts, a colony of foraging ants finds a worker allocation and transportation paths through the terrain that will optimize the colonies' consumption of these resources [4]. This allocation is found and continuously optimized through local interactions only, where each individual only reacts to the neighbors around it and learns and shares information with other members of the colony through deposited pheromone trails [7].

Since it can be argued that distributed coordination via local control has evolved in many biological instances, it makes sense to investigate how these strategies may be applied to the problem of cognitive radio coordination. This process of biomimicry, the analysis and adaptation of good solution concepts from nature to technical problem domains, has, in recent years, gained significant momentum as biologically inspired solutions have been shown to possess high levels of robustness and fault tolerance [17], while efficiently using available resources [26]. This study of a biological system and the process of adaptation to cognitive radio networks will be the focus of the following sections.

3.1 Requirements for concept transfer

While there exist many different strategies in nature for distributed coordination, it is important to find and select only such a strategy that directly fits the properties and challenges of the intended target domain. After an analysis of a variety of self-organization processes

in nature, it became apparent that, of those examined, only the swarm behavior as found in schools of fish and flocks of birds would provide an adequate solution to our problem of cognitive radio network coordination. This is due to three main abstract properties that are shared between the two domains:

The surrounding environment is memoryless Spectrum—while being a scarce resource—is inherently renewable. While most resources are depleted during consumption, radio spectrum is continuously available and does not change whether or not it has been previously utilized. This poses, however, a significant problem from the perspective of a coordination algorithm that tries to learn and pass on information about the environmental conditions or system state. Unlike an ant trail, which can persist over a long period of time, information passed on using a radio spectrum becomes unavailable as soon as it has been transmitted. While being rather abstract, this characteristic has a profound implication on whether a certain biological approach, such as an ant algorithm, may be used. The biological approach of flocking, however, shares this characteristic of a volatile environment (historical information is not preserved by the environment), thus making it transferable to the context of dynamic spectrum allocation.

State information can be directly observed on a limited horizon Even when a cognitive radio node has not configured itself to become a part of a cognitive radio network and actively exchanges information with neighboring nodes, it is still able to collect and infer information about its local information and the environmental conditions of those around it by overhearing the communication of its neighbors. This means that a limited amount of network state and node configuration information can be gathered without engaging in active and explicit communication. This is also the case for the approach of flocking, where the agents are able to directly sense the current position and trajectory of a limited number of neighboring individuals.

Individuals can communicate through actions Instead of relaying information about their changing environment to others through explicit communication, nodes may also pursue the opposite strategy and silently perform adaptive and corrective actions to recent changes important enough to require an action by the cognitive radio. These actions are observable by others, which may then react to them in turn. Consider the example when a remote station suddenly incurs an interference source that is corrupting its incoming and outgoing packets. Instead of notifying its neighbors

about this change explicitly (where the notifications might also become corrupted), the node could instead switch on forward-error correction and continue its normal communication. Remote stations observing this reaction may then in turn mirror the node's decision and restore and continue normal operation without the need to use and understand a specific protocol that encodes the environmental state. This communication-through-actions paradigm is also very central to the behavior of schooling fish and flocking birds: It is not necessary to see the danger yourself—but if some threat is great enough for a neighbor to start fleeing it would be wise for you to do so as well.

In our inspection, it became apparent that many biological approaches, while also performing a coordination task without global information, are actually unsuited for transfer and application to the problem of distributed configuration in a cognitive radio network. As only the biological approach of schooling and flocking shared the requirements and implicit assumptions of the cognitive radio network coordination task, we selected this particular algorithm for our problem of cognitive radio configuration. Since schooling fish and flocking birds use in principle the same algorithm (with the exception that flocking birds always need to maintain a minimum velocity for aerodynamic purposes), we will focus our discussion from now on specifically on the schooling fish. The following sections will now provide details of this application of swarm behavior, starting from a formalization of swarming, discussing the concept of semantic transfer and finally describing example adaptations in a cognitive radio network.

3.2 The mechanisms of swarm behavior

The collective and seemingly complex behavior in animal populations such as schools of fish and flocks of birds that is often referred to as “swarm intelligence” is actually the result of a set of simple rules that are followed by each individual in the population. Yet, when each member in the group observes these rules, everyone's individual behavior and the resulting interactions between the agents in the population create global patterns of self-organization, displaying properties beyond the capabilities of the single individual and the behavioral response encoded in the original simple rules. This global pattern formation is commonly referred to as *emergent behavior*, as it is only created through the interactions of many individuals working together and is not visible by looking at individual members or their individual, atomic actions in the group. In a school of fish, each individual is only aware of and reacting to

stimuli in its immediate environment—when placed in a group, however, the entire swarm displays a globally well orchestrated behavior, capable of guiding groups of one million fish in unison motion, avoiding obstacles and predators in a large, cohesive structure.

These emergent effects take place in two ways: First, while every member in the group follows only a number of rules populated with local information, the many interactions among the individuals give rise to a global dynamic reaction pattern. Second, although each individual is only aware of stimuli in its immediate surroundings, its own observation of the environment and the reactions of its direct neighbors, information, and observations propagate through the entire group, making all individuals and the swarm as a whole able to react to influences that may only have been observed by a few.

It is these two properties that make swarm behavior an appealing solution for the coordination of cognitive radio networks: The algorithm's ability to react globally to changes and information that only a few nodes sense locally provides a promising solution to the problem of global coordination as discussed in the last sections, while the simplicity of the rules minimizes computational overhead and removes the complexity of commonly used protocols resulting in high portability and applicability even on rudimentary hardware.

These rules that lead to coordination in schooling fish and flocking birds have been described in various literature (see, for example, [24]) and can be summarized as follows:

- *Cohesion*: Move towards other members of your group
- *Obstacle avoidance*: Keep a minimal safety distance from neighbors and obstacles
- *Alignment*: Match your direction and speed to those around you

Each of these rules, which are visualized in Fig. 2a–c, creates a certain component of swarming, and together, they result in the emergent behavior described above. Each subbehavior is activated dependent on distance as depicted in Fig. 3. At far distances r_c , members are attracted to other individuals in the group and by moving towards them larger and larger clusters are created, resulting in overall *cohesion* of the group. This tendency to group up is counter-balanced by the *obstacle avoidance* rule, which is only triggered at relatively small distances r_o . If individuals come too close, dropping below the minimum distance requirement r_o , the repulsion effect of this rule restores the safety distance, thus avoiding collisions between group members and other foreign objects. At mid-distances, the *alignment* rule

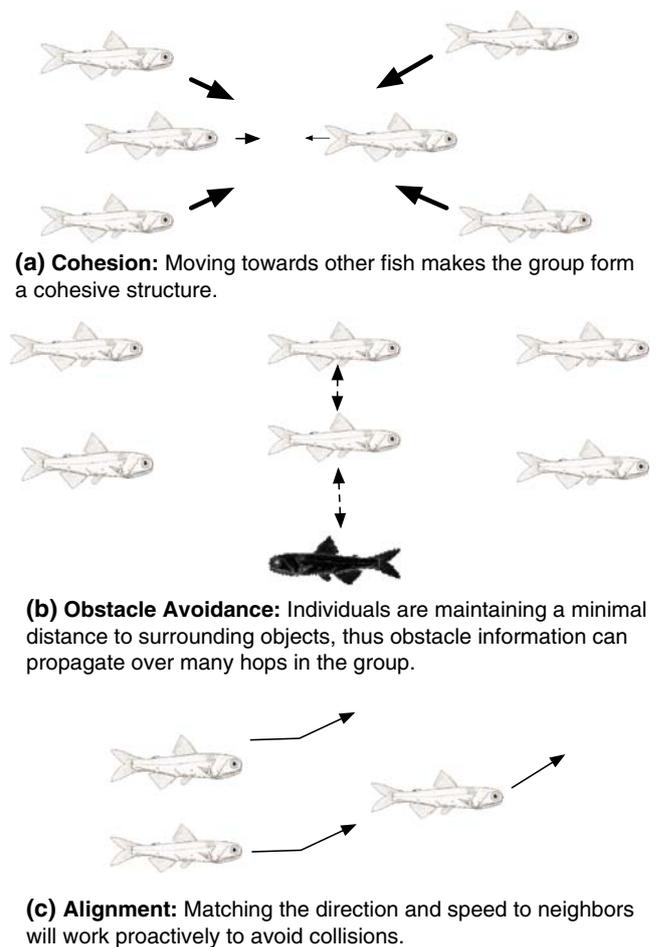


Fig. 2 The behavior of swarming is created through the application of three basic rules that are followed by every individual [11] (a–c)

adjusts the individuals' speed and direction to those around it in range r_a . This creates common movement among group members and minimizes the risk that the agents get either too close or out of range, thus stabilizing the group structure. The exact distances when the three subrules cohesion, alignment, and obstacle avoidance (sometimes also referred to as attraction, orientation, and repulsion) will be activated have been found to be species-dependent, but always follow the relationship $r_o < r_a < r_c$.

3.3 A formalization of swarm behavior

After the general operating principles of swarm behavior have been introduced, this section will now formalize these behaviors mathematically for further analysis. For the following analysis, assume the existence of several groups (as there could be several fish schools and later cognitive radio networks present at the same

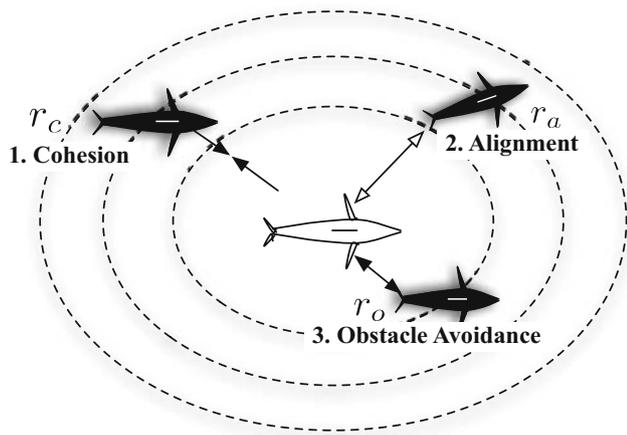


Fig. 3 The three components are activated only at certain minimal ranges and act as counterweights to create a dynamic equilibrium

time), denoted as S_i with $i = 1 \dots n$, each representing a school of fish or a set of obstacles. The groups themselves consist of a finite number of elements p_j^i with $j = 1 \dots |S_i|$. While the algorithm and the calculations are performed identically for each group and every element within them, for simplification of the discussion, we will provide the formalization of the swarm behavior rules from the perspective of element p_j in the group S_i .

Cohesion As discussed in the last section, the subrule of cohesion lets the members in the swarm group up to form and maintain one cohesive structure. This aggregation is achieved by each individual sensing and moving into the direction of other group members around it within cohesion range r_c . The individual can determine the direction to travel into in two ways: either by finding the movement vectors necessary to get to each neighbor within cohesion range and averaging the required movement vectors, or by averaging the current position of each remote peer and moving towards this average location. Both approaches will guide the individual to the same location, the perceived swarm’s center of mass.

While finding the center of mass is a task requiring global knowledge (the position of all individuals), the procedure exhibited by the cohesion rule in the swarm will approximate the same result using local information, provided that each member has at least one neighbor in its visibility and the patches of nodes are interconnected. How the cohesion rule creates a cluster at/near the center of mass can be understood intuitively in the following way: when an individual is located in the middle of a (sub)group, it will be surrounded by several neighbors on all sides. Therefore, its perceived center of mass (the average of its neighbors’

locations within range r_c) will be at or near its current position and the force to move to this average position is small. If a node is at the perimeter of a group or at a significant distance from it, most of the neighbors will be located at one side of the node. The resulting movement direction towards their average location will cluster the individuals and keep the group together.

Equation 1 formalizes this behavior. The direction that an individual has to move into according to the cohesion rule, denoted as \vec{v}_c , can be found through a vector addition of all movement vectors to the other members in the group. In the situation where each member has a limited input range r_c , it only considers those neighbors $p_k^i \in S_i$ for which $|\vec{p}_j^i \vec{p}_k^i| < r_c$.

$$\vec{v}_c = \frac{\sum_{k \in S_i} \vec{p}_j^i \vec{p}_k^i}{|S_i| - 1} \tag{1}$$

Obstacle avoidance Besides forming clusters and groups, it is also critical for each individual not to get too close to other objects but, rather, maintain a minimal safety distance from neighboring peers and potential obstacles such as members of other swarms or other foreign objects, thereby working as a counterweight to balance the contraction introduced by the cohesion rule.

Following this rule, each member tries to maintain and restore the minimal distance if a particular object gets within obstacle avoidance range r_o . This can be achieved by direct repulsion from such objects, a straight reverse, which is expressed by the negative direction vector from these entities to the node p_j ’s current position. If multiple objects trigger this rule and must all be avoided simultaneously, the resulting movement vector according to the avoidance rule, \vec{v}_o , can be derived similarly to the last rule by averaging the individual repulsion reactions.

Equation 2 formalizes the resulting avoidance behavior \vec{v}_o to avoid all neighbors and obstacles within collision range r_o . In order to define different priorities for violating foreign objects and members of its own group, assume C_s to contain all members in its own group that are too close and trigger the avoidance rule, while C_o contains members of other groups or foreign obstacles within avoidance range. For both of these sets, the sum of avoidance vectors as seen from p will be computed and, if necessary, weighted to yield the corresponding avoidance movement.

$$\vec{v}_o = \frac{-\sum_{k \in C_s} \vec{p}_j^i \vec{p}_k^i - \sum_{k \in C_o} \vec{p}_j^i \vec{p}_k^i}{|C_s \cup C_o|} \tag{2}$$

where $C_s = \{k \in S_i \mid |\overrightarrow{p_j^i p_k^i}| < r_o\}$ and $C_o = \{k \in (\bigcup_{\forall k|k \neq i} S_k) \mid |\overrightarrow{p_j^i p_k^i}| < r_o\}$

Alignment Although not immediately mandatory to achieve the emergent behavior of swarming, the subrule of alignment adjusts the current trajectories of nodes performing proactive and corrective actions, thereby preventing that the overall structure becomes too loose and potentially disconnected or too dense and requiring nodes to enter the collision avoidance mode.

This alignment can be calculated by subtracting the nodes' current trajectory $\overrightarrow{v_p}$ from that of all neighbors $\overrightarrow{v_k} \forall k \in S_i$ within alignment range r_a as formalized in Eq. 3. Since this movement vector contains information about both the angle and speed of movement, the resulting average will let p 's trajectory slowly converge to that of its neighbors.

$$\overrightarrow{v_a} = \frac{\sum_{k \in C_a} \overrightarrow{v_k} - \overrightarrow{v_p}}{|C_a|}, \tag{3}$$

where $C_a = \{k \in S_i \mid |\overrightarrow{p_j^i p_k^i}| < r_a\}$

To finally derive the overall movement of particle p for the next iteration, the movement vectors of the three subbehaviors are combined. While the individual movement components from the subrules may be simply added as vectors, weights may be introduced as indicated in Eq. 4 to achieve certain global properties.

$$\overrightarrow{v_p} = \overrightarrow{v_c} * w_c + \overrightarrow{v_o} * w_o + \overrightarrow{v_a} * w_a \tag{4}$$

These weights have been found to be species-specific [16] and may be used to fine-tune the emergent behavior responses of the overall group. Large weights for cohesion will, for example, let the group stay together as a single structure and avoid partitions as much as possible. Large weights for obstacle avoidance will let the group exhibit very timid behavior against outside obstacles.

3.4 How swarm behavior can be transferred to cognitive radio networks

Using the formalization of swarm behavior as presented in the previous section, it is now possible to use this concept towards the coordination tasks in cognitive radio networks. However, before applying these mechanisms of biological coordination to the technical domain of communication networks, it is necessary to step back and adjust the assumptions, concepts, and strategies of the algorithm to fit those of the target domain, i.e., one must translate the sensory input and

actions meaningful to a school of fish to those meaningful to a cognitive radio network. These differences and the abstract transfer process will be sketched in this section; a concrete example will follow in the next section.

As the mechanism of swarming behavior has evolved as a biological response strategy in various fish and bird species, it is clear that the approach is exactly tailored to the needs, senses, and actions of the organisms using this algorithm. Instead of directly applying the algorithm based on the formalization derived in the previous section and yielding a cognitive radio control algorithm that makes use of the cognitive radio's capabilities in a suboptimal way, it is necessary to make some changes to the swarm algorithm. This adjustment process is mandatory due to a variety of differences between the biological and technical domains:

Controlled variables As swarm behavior has evolved to form and maintain spacial structures among large animal populations, it naturally adjusts the physical location to achieve this task. While this is logical to control this variable in the source domain of the algorithm, controlling and adapting spatial position is typically undesired in the context of a cognitive radio network, where a device's position is usually predetermined or even fixed. Instead, the system should make use of other variables it can freely change, its adaptation capabilities, to achieve and maintain a good communication environment at a particular location.

Dimensionality In its original biological domain, each agent monitors and performs actions depending on its distance from surrounding neighbors. For a computation of distance, measured as the Euclidian distance in three-dimensional space, it is irrelevant whether a remote object's distance is greatest on the x, y, or z axis, as the three dimensions have identical semantics. In a cognitive radio system, however, a node may be able to adapt many different parameters, each representing a dimension along which distance may be calculated. These dimensions will, however, most likely have different interpretations, which can therefore not be directly interchanged and, thus, require a more complex calculation of the distance metric, i.e., reducing separation along dimension "transmit power" by three units does not improve overall reception if separation along the dimension "frequency" is increased by one unit.

Variable types and dynamics In addition to a different number and interpretation of dimensions, the type and behavior of variables may also significantly differ between the two different domains. While spatial

positioning and distance can be assessed using continuous variables, a cognitive radio system may use many parameters that can only be defined between certain lower and upper boundaries (for example, transmit power), exist only as integer variables (for example, frequency in a channel-based system), or are nonlinear (for example, antenna direction). These challenges must be reflected in the function calculating the equivalent of the Euclidian distance in swarming.

Despite these general differences, it is still possible to transfer and adapt the swarm behavior algorithm for the control of communication networks. For this step, and to overcome the differences discussed above, it is necessary to adjust the algorithm, its underlying assumptions, and strategies to those present in a wireless communication network and replace the sensory input and actions to allow proper operation in the context of a cognitive radio network.

One way to conduct these necessary adjustments is to translate the algorithm semantically, i.e., derive the abstract function that each subbehavior provides in nature and find an equivalent high-level function providing a similar objective for the cognitive radio network. This equivalent function is then populated with the sensory input and the possible adaptation parameters of the cognitive radio node.

From a high-level perspective, the underlying objective of a group of fish to form a school is to minimize the danger of an attack by predators [6], as individual members are more easily attacked than a group. Staying in the formation, therefore, offers protection, and the individuals use some of their potential free parameters (their ability to move within space) to achieve this objective.

At a similar high-level view, the objective that a cognitive radio network tries to achieve is quite similar. The cognitive radio network also attempts to maximize its resilience against negative outside influences to maintain good communication among its network nodes. Although physical proximity would certainly achieve this objective, a cognitive radio network has a multitude of other parameters it can adapt to shield itself from outside influences and strengthen its links to neighboring peers, for example, by modifying its transmission power, frequency setting, modulation scheme, or encoding parameters. The cognitive radio can then statistically analyze how each parameter setting and the resulting parameter interactions would affect the target variable it intends to improve, which could be efficiently learned through the use of fractional factorial designs [12]. If it has the objective to improve network communication by reducing the impact of outside interference, a cognitive radio could, for example,

increase its transmission power. Similarly, if equipped with a directional antenna, steering the receiver away from interference sources and into the direction of the peer would achieve the same result.

Similar mappings can be (automatically) found for the other parameters that the cognitive radio is capable to adapt and for all subbehaviors of the algorithm. This will ensure that, while the algorithm was modified to the parameters and input of the target domain, the general functioning is identical.

3.5 Example adaptations for a cognitive radio swarm algorithm

After the formalization of swarm behavior and a high-level description of the adaptation process, this section will highlight and discuss this adaptation process in detail. We will demonstrate the adaptation process using the exemplary cognitive radio parameters transmission power, antenna directionality, and frequency setting for the three subbehaviors cohesion, obstacle avoidance, and alignment.

Cohesion As discussed before, the abstract function of the subrule of cohesion is to initiate actions that will maximize the algorithm's overall objective. In the case of schooling fish, cohesion will move the individuals closer together, as increasing spatial density will provide more protection. Similarly, wireless devices should initiate actions that will maximize the strength of the communication links between them.

Consider again the formalization of the cohesion rule in the school of fish,

$$\vec{v}_c = \frac{\sum_{k \in S_i} \vec{p}_j^i \vec{p}_k^i}{|S_i| - 1},$$

which will urge the individual to the perceived center of mass of its surroundings. The position of an element j in swarm S_i , p_j^i , based on which the movement is calculated, is, for the case of the school of fish, a three-dimensional coordinate

$$p_j^i = \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

While, in the original domain of schooling fish, the current position of an individual is only determined by its locality, a cognitive radio system's "position," i.e., its current configuration, may contain much more information, which, however, may still be captured using this simple concept and its notation. Instead of using an identical semantics of the three dimensions as in the

schooling fish, we define each dimension of the cognitive radio’s “position” vector as one of its adjustable parameters. Thus, for a cognitive radio system that may set its transmission power, antenna directionality, and frequency setting, we obtain an analogous position of the system in the overall configuration space of

$$p_j^i = \begin{pmatrix} \text{value in configuration dimension 1} \\ \text{value in configuration dimension 2} \\ \text{value in configuration dimension 3} \end{pmatrix} = \begin{pmatrix} \text{transmission power} \\ \text{antenna directionality} \\ \text{frequency setting} \end{pmatrix}.$$

Since each of these components may now have different semantics, i.e., a simple Euclidian distance equivalent does not exist anymore, each distance metric now has to be computed individually for each dimension. Thus, for each dimension $D^{1..n}$ of the cognitive radio’s configuration space, we combine the observations made by local node i about the peer k using a combination function tailored to the underlying semantics of this configuration space $f^{1..n}$.

$$\vec{v}_c = \frac{1}{|S_i| - 1} \begin{pmatrix} \sum_{k \in S_i} f^1(D_i^1, D_k^1) \\ \sum_{k \in S_i} f^2(D_i^2, D_k^2) \\ \sum_{k \in S_i} f^3(D_i^3, D_k^3) \end{pmatrix} \tag{5}$$

The resulting cohesion “vector” will then, similarly to the original biological domain, determine how the cognitive radio will have to “move inside its configuration space” and adjust its communication parameters in such a way as to maintain and strengthen its communication link as much as possible. The remainder of this subsection will introduce the considerations behind these mapping functions $f^{1..n}$ for a selection of different cognitive radio parameters.

Transmission power One strategy for the cognitive radio to improve the fidelity of its communication links is to modify its transmission power until a sufficient link quality is reached. To demonstrate the coordination-without-explicit-communication paradigm, assume for simplicity a network with symmetric links where a link between nodes i and j has a link gain of g_{ij} . Each cognitive radio transmitter intends to send at a high enough power that the remote station will receive the signal at a sufficient signal strength. If P_{tx} denotes the transmitter’s current output power, the idealized received signal strength of $P_{tx} \cdot g_{ij}$ needs to be large enough to meet or exceed the minimum required signal strength P_{min} . The trivial solution would certainly be to

always transmit at the maximum power level, but this strategy will, on the one hand, unnecessarily deplete a mobile station’s battery and, on the other hand, lead to unnecessary interference levels in the network if all nodes follow the same strategy. Instead, the cohesion rule will increase its own transmission power P_{tx} until the incoming transmissions meet or exceed the minimum power level P_{min} , a process that, through the emergent behavior created in the repeated cycles of interactions, will lead to sufficient power levels across the network nodes. This parameter increase must, however, be counterbalanced and bounded, a function accomplished by the obstacle avoidance rule.

Antenna directionality Another strategy to improve the communication environment with peers is to adapt the cognitive radio’s antenna patterns if the node is equipped with a directional antenna. Through this modification, incoming interferences may be dampened as much as possible, while steering into the direction of the sender will improve the overall link gain. The less accurate the alignment of the directional antenna, the lower the received power level will be at the remote station. While the antenna gain would depend on the specifics of the used antenna, assume a hypothetical antenna with a cosine directionality gain for visualization of this concept. If \vec{dir}_{rx} denotes the receiving direction of the remote station and \vec{dir}_{tx} the transmitting direction of the local station, the offset of the two directions will determine how much energy will be received at the remote station. If the two stations are perfectly aligned, a full signal can be received, while an orthogonal arrangement will provide no reception. Using our hypothetical cosine antenna, this directionality factor will be the vector product of $\vec{dir}_{tx} \cdot \vec{dir}_{rx}$ (with a shift of $\pi/2$).

Frequency setting Finally, the fidelity of the received signal will also depend on the frequency selected to send and receive signals. Here, the cognitive radio may, for example, use the cohesion function to select a communication frequency exhibiting the lowest noise or interference level or providing the best data rate for the current application.

In contrast to the previously discussed parameters, however, the frequency selection of the cognitive radio is a binary variable. Assuming discrete, nonoverlapping channels, outgoing packets may only be received by remote stations if tuned in to the exact same channel. If the channels the cognitive radio node has at hand are overlapping or the node is capable of freely changing its transmission frequencies, the effect will be less drastic. In this context, the cohesion rule will drive the cognitive radio to select the frequency that would allow

for a maximization of the received signal strength and enabling communication with the highest number of peers.

These observations about the adaptations' effects on the cognitive radio's links can then be formalized in an analog way as Eq. 1, as shown in Eq. 6. With the characteristic behavior of the parameters transmission power, antenna directionality, and frequency setting as discussed above, the cohesion subbehavior using Eq. 6 will now attempt to increase the transmission power to the minimum level required, use a transmission frequency that will maximize the number of communication links with remote stations, and align the antenna direction, such that the maximal possible link strength is achieved. This optimization process may be conducted separately on a per-node basis, i.e., the cognitive radio would change, adjust, and maintain its parameters independently for different remote nodes.

Therefore, the combination functions used in Eq. 5 for a cognitive radio controlling these three parameters may be defined as

$$\begin{pmatrix} f^1 \\ f^2 \\ f^2 \end{pmatrix} = \begin{pmatrix} 0 \\ FR_i \\ \vec{\text{dir}}_{\text{tx}}(i) \end{pmatrix} \begin{pmatrix} P_{\text{tx}} * g_{ij} - P_{\text{min}} \\ FR_p \\ \vec{\text{dir}}_{\text{tx}}(p_i) \end{pmatrix} \quad \forall i \in S_i \quad (6)$$

Obstacle avoidance As discussed before, the abstract function of the obstacle avoidance rule is to counterbalance the effects of the cohesion rule and to make the swarm responsive to the negative impact of outside influences. In the case of the schooling fish, the obstacle avoidance rule maintains a minimum distance from all objects, as too much proximity may prove to be harmful. In the case of the cognitive radio network, the obstacle avoidance rule may be used similarly to make the network responsive to interference and to bound and limit the growth of certain parameters if too-high levels will have negative side effects, as for the case of transmit power. We will therefore consider these two application cases of the collision avoidance rule separately, first its usage to avoid outside influences, and second its usage to counterweight the cognitive radio's parameters.

Avoiding outside influences In its deployment context, a cognitive radio network will be exposed to some amount of interference by primary and secondary users, contributing to the overall noise floor experienced on each channel. It is the cognitive radio's objective to minimize the amount of incoming interference, an objective that may be trivially achieved by adding a term to the obstacle avoidance function that incorporates the current noise floor level of each channel. This term will then later—after a combination with the other two

subbehaviors—balance the frequency selection, so that a frequency is chosen maximizing the number of links to other nodes while simultaneously minimizing the amount of local interference.

Providing a counterweight In addition to creating timid reactions to outside influences, the obstacle avoidance rule also acts as a counterweight against the amplifying actions of the cohesion rule. As discussed above, the cohesion rule will strive to improve the communication links, which may result in some parameters being continuously increased to a maximum value, eventually hurting the performance of the network as a whole. To avoid this situation, the terms in the collision avoidance rule add a penalty punishing overshoot targets, and thereby limit such parameter values to the minimum possible value that still meets the underlying objective. In the example of transmission power introduced above, the power setting of the sender should be chosen in such a way that the minimum required signal strength P_{min} is met or exceeded by a small difference. The corresponding term in the collision avoidance rule would therefore add a penalty for each unit that this value is exceeded, thereby self-regulating the system to maintain a sufficient, but reasonable, setting. Similarly, other counterweights must be introduced for all parameters used in the cohesion rule that might be subject to self-amplification or if certain minimum/maximum levels must be maintained at all times. These considerations and examples for the collision avoidance rule in a cognitive radio are modeled in Eq. 7.

$$\frac{-\sum_{i \in S_i} (P_{\text{tx}} * g_{ij} - P_{\text{min}}) - \sum_{i \in C_o} P_{\text{tx}}}{|S_i \cup C_o|}, \quad (7)$$

where $C_o = \{k \in (\bigcup_{\forall k|k \neq i} S_k) \mid |\vec{p}_j^i \vec{p}_k^i| < r_o \ \& \ (FR_k == FR_j)\}$, and FR_i denotes the frequency of node i .

Alignment As the underlying, abstract function of alignment is to provide proactive and corrective actions to the swarm's behavior before (de)activations of the other two rules become necessary, a similar stabilization feature may be added to the cognitive radio rule set. While alignment is not mandatory to the proper functioning of swarm behavior, it aids the algorithm to respond more fluidly to abrupt changes. If applied to the context of cognitive radio networks, such functionality may be introduced on a per-term basis for the parameters used in the cohesion and observation avoidance rules. Generating such alignment terms may, however, be difficult, if not impossible, for system parameters that are discrete or measured

in discrete time steps and, therefore, do not have any intermediate values. In these situations, it is possible to replace the alignment rule for these parameters by a trend prognosis, thereby obtaining the ability to proactively react to potentially upcoming future changes.

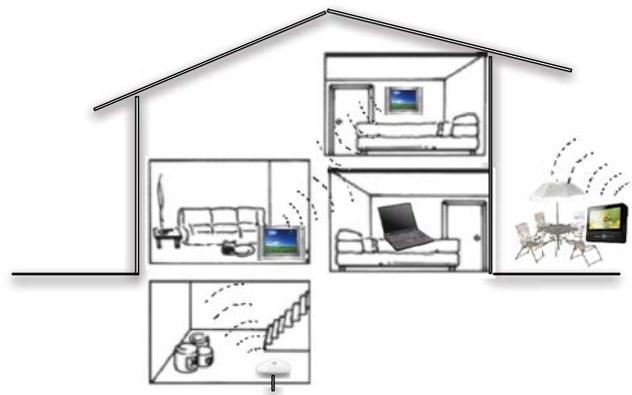
4 Repulsion and attraction in a cognitive radio network

After the discussion of the general biological concept of swarming and a presentation of how such a biological algorithm can be transferred and adapted to the domain of communication networks, this section presents a specific cognitive radio swarm model that was implemented in both simulation and as a hardware deployment. In addition to this exploration, this section also focuses on certain aspects not present in the biological algorithm, i.e., requirements and limitations that are unique to technical hardware and that, therefore, need to be integrated into the algorithm itself.

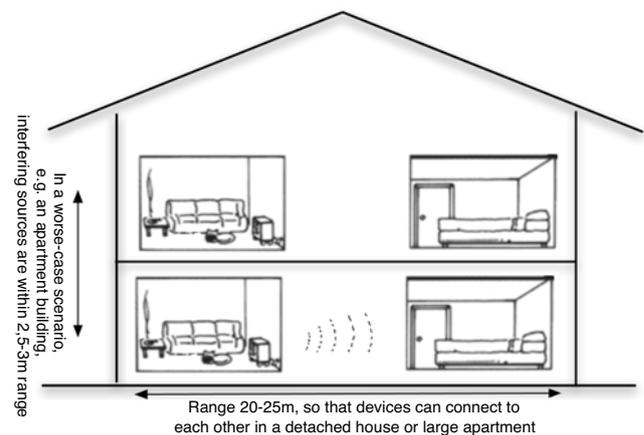
4.1 The dynamic configuration of wireless home appliances—a cognitive radio reference scenario

Before going into an in-detail discussion of the application of the swarm algorithm in a cognitive radio network, this section will first discuss the evaluation scenario used for our study. The evaluation specifications were derived from a set of four possible cognitive radio system tasks that will be required in a potential application context for a cognitive radio—the dynamic configuration of wireless home appliances as presented in [8]. Figure 4 shows this potential application context: the pervasive use of home appliances (such as home entertainment systems) meshing up their services and exchanging digital content wirelessly. In this scenario, appliances such as TV sets, DVD players, and home stereos would exchange information about their capabilities and make their services available to each other (see Fig. 4a). The home stereo could, for example, advertise its capability to play audio at high fidelity to the TV, which then automatically can stream the music of a movie wirelessly to the stereo for playback. Ideally, these appliances would have sufficient range to mesh up with any other device in a house or large apartment. Figure 4b shows the downside of such a pervasive deployment, as in an apartment building sufficient range to connect to devices *inside* a residence will introduce amounts of interference to devices *outside* in other residential units.

Cognitive radio technology would be beneficial to these deployments in two ways: First, it would provide a way of self-configuring the appliances to automatically



(a) The appliances in a user's deployments independently configure themselves to a common control channel or per session channel assignment,



(b) while avoiding interference from surrounding installations.

Fig. 4 The dynamic coordination and configuration of wireless home application is a potential application scenario of cognitive radio control algorithms and will be used to derive a benchmark and metric for the evaluation of our proposed algorithm [8] (a, b)

connect and work together, selecting appropriate frequencies to exchange content. Second, it would enable the devices to avoid interference from outside sources and maintain appropriate QoS through adaptation. If we assume a cognitive radio system to be deployed as the driving technology in such an application context, we can derive four key tasks that will need to be performed by the control algorithm coordinating the network nodes:

Common channel assignment Many networking and cognitive radio algorithms are built on the assumption that a commonly available control channel is available across the network and may be used for various communication management tasks. Unless placed on a dedicated (and potentially even licensed) band, such a channel may not be predetermined by the network

nodes and may have to be determined at runtime. In addition to the swarm algorithm's ability to configure the cognitive radio network nodes to a common configuration, thus allowing communication to take place, the algorithm could also be used as a bootstrapping procedure to preconfigure such alternative (cognitive radio) networking algorithms, thus allowing such proposals to be deployed even in very dynamic environments.

Opportunistic reuse of spectrum Following its objective to minimize interference on a per-flow basis, the algorithm is allocating and spreading out the network's data flows in such a way that interference to other nodes is minimized as much as possible. In a typical application scenario, however, a particular flow will not completely utilize all resources provided by a given channel, so that additional possibilities for opportunistic reuse of spectrum, even within the swarm algorithm's solution, are created. In this abstract scenario, it will therefore be investigated whether the algorithm can detect such unused resource allocations and dynamically reclaim them for the servicing of other network flows.

Spectrum-aware routing Due to spectrum heterogeneity, it will often be the case that flows within a network may not be directly serviceable, as a common communication frequency between sender and destination does not exist, but must be accommodated using spectrum-aware routing where intermediate nodes are located that share both a link and a common communication frequency and, therefore, forward network data flows. A special case of this spectrum-aware routing is the common channel assignment problem in the case of spectrum heterogeneity where no single common channel exists but only certain areas can settle on an area-wide piecewise assignment with stations at area borders relaying traffic flows.

Multiflow routing The last application scenario will determine how well the swarm algorithm makes use of cognitive radios equipped with multiple orthogonal wireless interfaces. Whenever there is enough available spectrum that is not being used by any other service and there is also a high enough flow demand within the network, it makes sense to opportunistically use as many channels as possible, i.e., as many orthogonal frequencies as the sender, receiver, and possible intermediate relays share and can simultaneously access, so that the amount of spectrum claimed by each node is determined dynamically with respect to the available spectrum and current internal and environmental traffic conditions, and flows can therefore be accommodated as quickly as possible.

While we have evaluated our proposed algorithm against the best possible reference solution as obtained through a mathematical programming approach in all four cognitive radio application tasks, in this paper, we will limit our discussion to results obtained in the first and third evaluation tasks only and omit further details to the other configuration assignments due to space constraints. It may, however, be noted that, in order to address these tasks, the cognitive radio algorithm does not require significant modification, only an addition of the relevant terms to the cohesion, obstacle avoidance, and alignment that would let the control algorithm, for example, also consider dynamic reuse or routing information in its decision-making process. Details about the mathematical formalization of these reference tasks based on the multicommodity flow problem are discussed in detail in [10].

4.2 A cognitive radio swarm model

After the evaluation scenario has been specified, this section will provide details on the actual implementation used in the deployment to achieve the first configuration task. Using the approach presented in the last few sections, we implemented a cognitive radio control algorithm based on the principles of swarming to distributively manage the radios' configurations. While such an algorithm could use a variety of different parameters for the adaptation process, this work—in order to explore the feasibility of the general concept and to maintain a tractable performance analysis—will focus only on a frequency adaptation model through swarming and leave the configuration of the other parameters open to different algorithms within a cognitive radio system.

The control algorithm's main objective in the framework of swarm behavior is to achieve the lowest possible interference to the local radio and the network as a whole. This interference is measured in terms of background noise and interference generated by other stations not being part of the cognitive radio network at hand. While avoiding interference to allow for the best possible link fidelity, the control algorithm must also consider the remote station's spectrum availability and the current network flows that are either generated, received, or routed by the local station at hand. Thus, the swarm algorithm's objective can be summarized as follows: select the local transmitter frequency/frequencies in such a way that the interference experienced by the local station is minimal while maintaining a feasible link connection to all remote stations in the network in which the radio is currently communicating.

Although both hardware and software deployment are built on the exact same model of swarming as discussed in the previous section and use the exact same objective function, some implementation specifics are disjointed enough to make it worthwhile to explain each implementation part separately such that sufficient detail unique to the particular component can be understood.

4.2.1 Hardware testbed using the Atheros wireless chipset

Due to economic considerations, we implemented the cognitive radio swarm algorithm on a hardware testbed platform, using a set of off-the-shelf Dell Inspiron laptops. Each laptop was equipped with a built-in wireless network interface based on the Atheros wireless chipset, which provided us, in combination with the Madwifi [2] driver, with a low-cost, low-end, software-defined radio system.

Each system could optionally be equipped with a second Atheros-based wireless interface and, in order to provide additional limited sensing capabilities, with a WiSpy spectrum scanner [3], a low-cost noise floor measurement unit for the IEEE 802.11 frequency band. A custom written driver for the WiSpy device was able to provide the cognitive radio with a just-in-time view of the 802.11 band's background noise at a 1-MHz granularity and 2-ms resolution rate.

The two devices were directly addressed by the swarm control algorithm, which was implemented as a separate structure aside from the regular network protocol stack. This system architecture was deliberately chosen to minimize software maintenance expenses and to provide for a maximal software portability of the system.

Figure 5 shows the architectural overview of the system. The wireless interface connected to an external antenna is controlled by the Madwifi driver, which exports a network interface towards the Linux operating system. This network interface is then used by the network protocol stack to receive and send packets. The Madwifi driver in this setup additionally exported a second network interface, which captured and forwarded all received and decodable packets (not only those intended for the local receiver) to the control algorithm for traffic monitoring purposes. This so-called monitor interface allows the cognitive radio control algorithm to overhear all data on the channel and, thus, obtain a passive view of the stations that are present and transmitting on a particular frequency.

Since the wireless network interface connected to the network protocol stack and control algorithm is

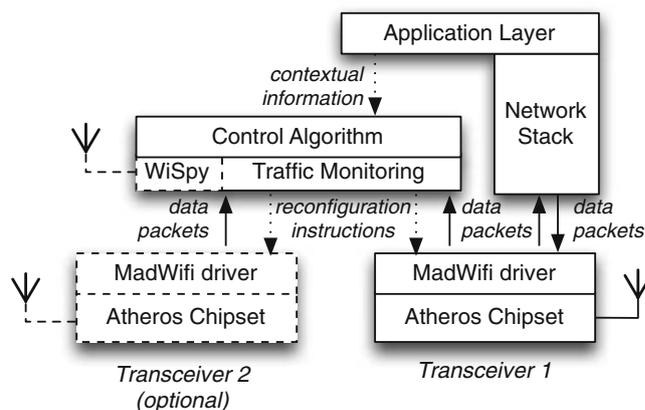


Fig. 5 System architecture of the control algorithm's implementation on a commodity hardware platform

being used for the cognitive radio's communication with other nodes, it will, most of the time, be set to the frequency currently chosen for transmission, as switching to different frequencies for scanning purposes will create some switching delay during which the node will be temporarily unable to receive incoming packets. To overcome this difficulty, the control algorithm also had the opportunity to manage and listen to an additional, optional wireless network interface. When equipped with such an additional interface card, the primary interface could be completely devoted to receiving and sending payload traffic while the secondary device would only be used for traffic monitoring. In addition to monitoring these network interface devices, the control algorithm could also optionally be connected to a WiSpy spectrum scanner and provide hooks to the application layer of the cognitive radio node through which it was able to learn contextual information about the network topology (for example, to which remote stations the current application wants to establish and maintain links and which nodes are not used by the current application). After reading in all these system and environmental information as input, the control algorithm then performs a frequency selection using the swarm algorithm discussed in the previous section and configures the primary (and if available, the secondary) network card's frequency accordingly.

To save resources, the control algorithm was only invoked in certain intervals. This interval length is freely configurable and should, in practice, be selected to match the volatility of the environment and the system's desired response in the current application context. Even though the control algorithm remains in between these invocations in sleep mode, it is still acquiring information about its environment during this idle period as the input streams from the network

interfaces, application layer, and WiSpy are buffering incoming status information, packets, and sensory data. In order to allow for an analysis of the algorithm's behavior and its adaptation decisions at a fine granularity, we set the control algorithm to be invoked in only 1-s intervals.

In order to facilitate a variety of different experiments in the hardware testbed and due to the difficulty to manage the availability of physical links, i.e., control which node is able to observe and communicate with another peer, network topologies were virtually created and managed through the application layer's interface with the control algorithm. This allowed us to create arbitrary topology setups in our experiments.

4.2.2 Emulation using a software simulator

In order to be able to test the control algorithm in larger and more complex network topologies, the control algorithm was also implemented and evaluated in a software simulation environment. The main focus of this simulation implementation was to study the communication patterns and interactions between nodes and to determine the convergence and connectivity of larger cognitive radio networks using the swarm behavior algorithm. The simulator was written as a discrete-event simulator in Java, building upon the common Lee propagation model [20]. Special care was taken to create efficient simulation kernel routines and good integration with high-cycle-length random number generators in order to avoid correlation issues often present in simulation environments [18].

To allow for large-scale testing, the simulation environment was enhanced by two additional components: first, a parameterized automated topology generator and, second, an automated test environment. The first component automatically created large batches of network topologies, which the control algorithm would be evaluated in. This generation could be parameterized, so that constraints on the node location, minimum/maximum node degree, interference sources, channel availability, and spectrum heterogeneity would be present in a given topology instance. This ensured that the network topologies being tested had sufficiently dense areas, as would be expected in a real-world deployment, but also required the algorithm to cope with several-hop links, a requirement identified by networking researchers to be a critical condition for network algorithm testing [19], as the fidelity of network management and routing algorithms is only properly tested in networks with a sufficient degree of average hop counts. These topologies (typically between 1,000 and 10,000 for a given parameter combi-

nation of network size, node degrees, etc.) were then provided by the second component to the simulation environment. After the simulation, the interaction and response patterns of the swarm behavior control algorithm were statistically analyzed.

4.3 Simulations using the QualNet simulation environment

After the general applicability of the algorithm in larger network sizes had been confirmed using the interaction-based simulator in the last section, the control algorithm was additionally tested in a second simulation environment to validate the results of the previous evaluations and to further investigate the effect of certain parameters, such as the algorithm's ability to scale to large amounts of spectrum and to study the effect and benefits of an external utility such as the WiSpy energy sensor.

For this part of the evaluation, the swarm algorithm as discussed in the previous section was implemented within QualNet, as this simulation environment is commonly used in networking research, provides good support for wireless modeling, and offers high scriptability so that a multitude of different scenarios created by the previously discussed topology generator could be evaluated. The swarm behavior control algorithm was implemented from scratch using the simulator's built-in 802.11 PHY and CSMA/CD MAC modules.

4.4 Technical limitations and their implications on the biological algorithm

As discussed in Section 3.4, there exist a number of differences between the biological source and technical target domain, which may make the adaptation and application of the algorithm in the new context difficult. As previously explained, one of these differences is the discrete nature of some variables, such as "frequency," in a cognitive radio system, which makes the calculation of a distance metric for cohesion and obstacle avoidance or the calculation of a gradient for alignment challenging. For this particular variable, there exists an additional complication resulting from technical limitations of the hardware. To address such issues, depending on the type and severity, it might be necessary to modify and enhance the biological algorithm to also allow for convergence in the presence of this and similar hardware limitations. This section discusses how such limitations may be overcome.

Due to design choices and economical and technological concerns, the capabilities of the RF components in most transceivers are fairly limited, so that such a

system is, for example, not able to process signals from a large number of frequencies at the same time. The same is true for the 802.11 commodity components used in our hardware testbed, which, due to their typical use case, would be set and operated on a single frequency. Future software-defined radios will certainly provide broader capabilities, but these capabilities will also come at a higher system cost. It is therefore worthwhile to investigate how it is possible to enhance the biological algorithm to be able to cope with such technical limitations and allow for fast convergence also on rudimentary hardware.

As we are using commodity WiFi components for our hardware deployment, our nodes are limited to the 11 channels of the 802.11 band. In addition to the limitation to these channels, the wireless interface card can only be tuned to a single frequency at a time. As noted in the previous literature, spill-over effects make it possible to also receive packets from adjacent channels, thus enabling the radio to monitor the activity on a few channels at the same time.

Figure 6 shows the problem created by this limited scanning range in the context of the biological analogy of swarming. If we imagine running the algorithm in the 802.11/ISM band, with a total of 11 channels available and the radio able to monitor five frequencies at a time (current transmit frequency ± 2 channels), a certain amount of activity in the cognitive radio's neighborhood will go undetected. If these activities take place on frequencies other than the five monitored, these events will not be considered by the cognitive radio control algorithm and convergence might therefore be slower than if the algorithm could scan all available frequencies at once. This implication becomes visible if we translate this 5/11 ratio back to the original domain of swarming in schools of fish. A 5/11 ratio of visibility

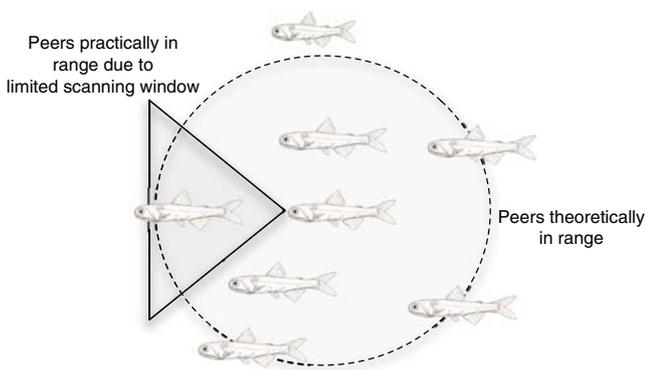


Fig. 6 In case the hardware is not able to scan the environment completely, the original biological algorithm might not converge [13]

to a fish would mean that each individual is only able to see and detect other group members in a 160° viewing window, as indicated in the figure. This would, however, mean that many neighbors present in cohesion and obstacle avoidance range would go undetected by an individual, preventing it from reacting to the presence, and thus, severely limiting the performance of the swarm.

If we follow this biological analogy of a 160° viewing window as shown in Fig. 6, it is easy to identify the problems that may occur in this situation:

- There exist peers in scanning range but outside the scanning window. The individual, therefore, does not consider these hidden neighbors and does not align with them.
- Group members might be temporarily within the scanning window but, due to movements, leave this window and are, therefore, also lost for the decision-making process.
- The individual might be within the scanning window of other peers, but they do not join up with the individual as initiated by the cohesion rule because they see other items in their windows that prohibit an approach due to the triggering of the obstacle avoidance rule.

Using this background of the biological analogy, it is straight-forward to design remedies that would address the issues of limited scanning windows in a group of schooling fish, and which can also address the convergence issues in a cognitive radio network. The remainder of this section will discuss three additional components to the cohesion and obstacle avoidance rules, which allow the algorithm to converge rapidly even when only possessing a partial view of the environment.

The first problem of nonsynchronization when peers are outside the scanning window can be easily solved by introducing another component to the cohesion rule. In periodic intervals or when a node does not sense any other node, it performs a sweeping scan of the spectrum and joins the largest aggregation of nodes on a particular frequency if that channel is locally available. This action, which would be equivalent to a periodic 360° turn, will ensure that nodes with limited knowledge of the environment will group up into a connected network.

The second problem of temporary visibility can be solved in a similarly intuitive way. Consider the situation where a neighbor leaves the fish's angle of view and does not return. A logical reaction in the biological analogy would be to briefly turn around and look into the direction in which the peer was headed. If there

exists another, larger group than the fish is currently part of, it joins as well.

That these intuitive rules would also address potential convergence problems due to limited sensing in cognitive radio networks is shown in Fig. 7a. After a series of adaptation decisions, the nodes A, B, C, and D settle on frequency 4, whereas nodes F and G detected each other on frequency 9. Since neither the first nor the second group can see the others due to the limited scanning window (channel $4 + / - 2$ and channel $9 + / - 2$), both groups stay on their corresponding channels and, therefore, do not form a joint network structure.

Consider now that node E is in range of peers D, F, and G and, therefore, has links to both subnetworks. If E would execute the first additional convergence rule, i.e., scan all frequencies in corresponding intervals, it would first see and announce its presence on channel 4, thus, briefly joining the first subnetwork. On its way to complete the entire environmental scan, it would then come across the second subnetwork on channel 9, where it would detect the presence of other nodes and also be detected by them. Since, from the perspective of node E, channel 9 is the better option (two network nodes instead of one), it will stay and permanently associate with F and G. After a while, D notices that

node E (which temporarily associated with it) left and did not come back, and executing the second additional convergence rule, now also scans the other frequencies to determine the new channel of that node. Once it detects the other cluster, it will join these nodes, if possible, and a corresponding scan-and-join action will propagate based on the same rule through the first subnetwork, thus creating a cohesive network structure despite the limited sensing capabilities.

The third issue of remote peers not joining up with a local node because they see obstacles that the local node does not see can also be solved through an intuitive additional rule. If, in the school of fish, the remote peer is not joining the local cluster, the local individual can infer that, from the perspective of the remote group member, its respective position is superior to the local node's situation and, therefore, does not join. While this may or may not be the case and cannot be decided definitely by an individual due to the limited scanning capabilities, if this static situation persists for a certain amount of time, either one should join the other group member on a trial basis as it can then determine if the remote situation is indeed superior and, in that case, stay there, which will trigger the remaining cluster to mirror that decision.

Figure 7b depicts how such a situation translates to a cognitive radio network. Nodes A, B, and C have converged on frequency 4, and nodes D, E, and F have settled on frequency 5. While node C detects node D on an adjacent frequency, it does not join this remote node because it also senses the influence jammer J1 on channel 5, which, from its local perspective, would make the remote configuration an inferior choice compared to its current configuration, and therefore, does not join the other subnetwork. After a certain amount of time in this static, suboptimal situation has passed, either of the two nodes (C or D) decides to try out the remote station's selection. If the selection turns out to be better, it will stay there; otherwise, it will return to its original setting. In this example, node D tries out channel 4 before node C makes an action. Node D will notice that it will receive additional benefits by being able to connect to the other nodes A, B, and C and stay there, and the remainder of its subnetwork will mirror this decision, thus, merging the two subnetworks into a cohesive structure.

Through the addition of these rules, it is possible to maintain convergence using the swarm algorithm for both the biological original and the technical adaptation of this concept in the presence of limited sensing capabilities. It is worth noting at this point, however, that the reason why such rules have not evolved in the biological domain and are now artificially introduced

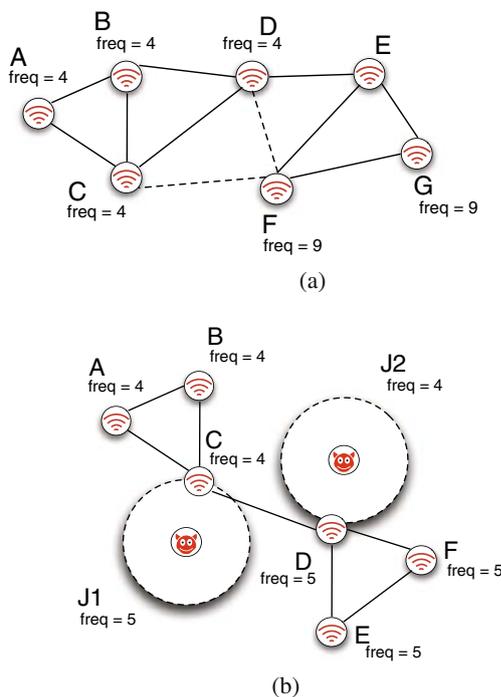


Fig. 7 Fast convergence is maintained in the presence of limited scanning capabilities when additional rules are specified [13] (a, b)

Table 1 Variables and their definitions used within the framework implementation of the local control algorithm

| Variable | Description |
|--------------------------|---|
| O_i | Classes of observations containing one or more actual time-stamped observations. Each class is obtained from one particular type of sensor. |
| $f_i(o \in O_i, k, [j])$ | Evaluation functions f_i that assigns a value to each observation of O_i in the dimension k , j (j can be optional). |
| D_i | Matrices of processed sensory input, one vector for each class of observations available to the system. The dimensionality is dependent on the number of aspects in which the data are evaluated. |
| w_i | Vectors of scalars weighting the relative importance of each dimension of D_i , i.e., $\text{Score}(O_i) = w_i \cdot D_i$. |
| ω_i | Weighting factors for combination of information obtained from sensors. |
| $m(\text{List } D_i)$ | Mapping function matching the set of observation classes. |

into the algorithm, is that animals engaging in swarming have 360° vision and, for their context, therefore, have a complete view of their surrounding environment.

4.5 Algorithm

After the discussion of the transfer process to the specifics of a cognitive radio network, this section provides a formal algorithmic listing of the value functions used in the pure cohesion and obstacle avoidance rules for the first reference task as implemented in the tested nodes and the software simulation environments.

Table 1 lists the variables and their definitions used within this framework. The algorithm bases its decision making on observations obtained from its internal and external sensors. Each observation is time-stamped and grouped into classes O_1, O_2, \dots, O_n containing only homogeneous information collected from one sensor. Since the observations collected by the cognitive radio's sensors can be analyzed in a variety of ways, each class of observations O_i is assigned a scoring matrix D_i , containing scalars $D_i(1, 1), D_i(2, 1), \dots, D_i(n, n)$ for each disjoint way in which the data collected by the specific sensor i are analyzed. To summarize the data from each type of sensor, there is also a weighting matrix w_i assigning each dimension of the scores D_i a relative importance. The weights in this matrix may be dynamic. For simplification, it will be assumed that the local node has index 0.

While this approach can be repeated for all evaluation functions used in the four reference tasks, we limit our discussion in this paper to an exemplary listing for the first situation of common channel assignment in spectrum homogeneity.

For this task, the cognitive radios collected two types of environmental information. The observations categorized as O_1 are information about overheard wireless transmissions and contain MAC address, receive time, signal strength, and the channel on which the transmission was received. If the cognitive radio is also equipped

with the optional WiSpy wideband energy detector, O_2 contains samples collected by this sensor.

Each observation collected by the sensor O_1 is analyzed by the value function shown in Fig. 8. This function assigns scores to each overheard transmission,

Algorithm 1 Determine cognitive radio parameter selection

```

1:  $S \leftarrow 0$  {scoring vector for the cognitive radio's parameter values}
2: for all  $O_i \in \{O_1, \dots, O_n\}$  do
3:    $D_i \leftarrow 0$ 
4:   for all  $o \in \{O_i\}$  do
5:     read signature of  $f_i$ , determine possible parameter combinations of  $k[, j]$ 
6:     for all  $k[, j]$  do
7:       evaluate  $f_i(o, k, [j]) \rightarrow$  update  $D_i$  {use sensor information to update system knowledge}
8:     end for
9:   end for
10:   $D_i \leftarrow w_i \cdot D_i$  {weight dimensions of sensor data}
11: end for
12:  $S \leftarrow m(\{D_1, \dots, D_n\})$  {call application-specific mapping function to combine sensor data from various sources}
13:  $S_m \leftarrow \max(S)$  {Choose the maximal scoring parameter value and adapt the radio correspondingly}

14: update radio demands
15: if radio can select several choices in parallel then
16:   repeat
17:      $S_m \leftarrow \text{next largest}(S)$ 
18:      $\text{adapt}(S_m)$ 
19:     update radio demands
20:   until radio demand = fulfilled or all  $S \leq 0$  or radio's parameter choices are exhausted
21: end if

```

$$f_1(o, i) = \begin{cases} 0 & \text{if signal strength is below consideration threshold,} \\ & \text{or more recent report of station exists in } O_1, \text{ or channel}(o) \neq i \\ \text{signal_strength}(o) & \text{if MAC address is part of subnet,} \\ -\text{signal_strength}(o) & \text{if MAC address is not part of subnet.} \end{cases}$$

Fig. 8 By scoring channels based on overheard incoming transmissions, the value function recommends a channel that would maximize the number of potential links to neighbors

a positive value proportional to the received signal strength if the node belongs to its own network, a negative value if it is part of a competing deployment, and zero if the observation is outdated or the incoming transmission is below a certain threshold value.

Correspondingly, there exists another value function that is analyzing the observation O_2 collected by the optional WiSpy sensor:

$$f_2(o, i) = \begin{cases} -\text{noise_floor}(o) & \text{if (ch}(o) == i) \\ 0 & \text{otherwise} \end{cases}$$

Here, all previous noise floor samples from each monitored channel are combined to receive a long-term channel assessment.

Since the incoming sensor data will need to be analyzed on a per-channel basis, the resulting aggregate assessments will be placed in a matrix large enough to contain information for each channel. If certain dimensions of the matrix, which, for this task, equal the channels, are more preferable than others (for example, by a particular channel having a lower licensing cost or offering a higher capacity), this preference can be expressed through the weighting vectors. Otherwise, the weighting vectors are set to be unity vectors.

To analyze the aggregated information, the corresponding value functions will be used to evaluate the collected sensor data. Since, in this situation, both general noise floor and signal strength are assumed to be equally important, the two vectors will be simply added arithmetically by the mapping function, which is possible since both vectors are of the same size and semantics.

$$D_1(i) = \sum_{o \in O_1} f_1(o, i) \quad \forall i \in [0, \# \text{ channels}]$$

$$D_2(i) = \sum_{o \in O_2} f_2(o, i) \quad \forall i \in [0, \# \text{ channels}]$$

The largest component returned from the mapping function will indicate the best configuration that currently should be selected.

5 Evaluation and results

In this section, we evaluate our proposed solution to cognitive radio coordination and discuss performance results obtained from an implementation of the swarm-behavior-based control algorithm in a hardware testbed and in two simulation environments. This evaluation is done in several parts: *First*, we show in a proof of concept using an experimental trace collected from the hardware testbed how the local decision-making process leads to global coordination. *Second*, we present performance results on the configuration efficiency of the local control algorithm for the reference tasks “common channel assignment” and “spectrum-aware routing” from the presented cognitive radio reference scenario, and compare them to the theoretically possible optimum. *Third*, since a cognitive radio will likely be deployed in a heterogeneous environment, we discuss the swarm algorithm’s ability to coordinate with other, potentially incompatible control algorithms and present coordination results obtained from the hardware testbed. *Fourth*, we study the algorithm’s dependency on internal and external factors, in particular the influence of the node-activity levels on the configuration success and the scalability of the approach as the total amount of the available spectrum grows. *Fifth*, we investigate to what extent a cognitive radio using the swarm algorithm may be enhanced through external knowledge, in our case, the addition of a low-cost spectrum scanner.

5.1 Coordination without communication: a proof of concept

The first part of our evaluation demonstrates through a simple proof of concept that coordination without communication in a cognitive radio is feasible and that a control algorithm based on local observation and reasoning can, thus, be used to configure a cognitive radio network in a decentralized manner. For this concept test, we deployed the cognitive radio algorithm as described in the previous section on a set of commodity hardware nodes (see Section 4.2.1), each possessing only one wireless interface that could be used for

scanning or exchanging communication. The hardware testbed was set up in such a way that there was full connectivity between the network nodes; however, one node was able to sense some external interference. This interference was introduced by an Agilent signal generator, which generated 802.11 frames not following the timing and back-off schemes in the specification thereby monopolizing the channel. Furthermore, the interference source was strong enough to significantly reduce and (together with its malbehavior) prohibit any communication within the network and, therefore, had to be avoided.

As this interference source was only detected by a single node in the network and, during the experiments, the source of interference was always placed on the channel that this node had chosen for communication, the impacted node could not inform the other members of the network about the presence of the interference source and ask them to reconfigure to an alternative frequency, as the channel immediately became unusable. Instead, it could only avoid the interference source itself by switching to a different channel and waiting for the other network nodes to rendezvous on this new channel due to the rules of the swarm algorithm.

This procedure and the rendezvous behavior created by the swarm algorithm is shown in Fig. 9, which displays the state of the network environment from the view of the node exposed to the interference (node 1). The figure shows the spectrum noise level measurements for 30 MHz of the IEEE 802.11 band over a period of time¹ where brighter colors refer to a higher energy intensity on a particular frequency. The figure clearly shows how, over the course of the experiment, the signal generator switches from channel 2 to channel 1 where the network had currently converged to. The colored dots indicate the time and frequency when node 1 receives a data packet from a remote station (node 2's packets are colored blue, node 3's are red). As the nodes possess only one interface for scanning the spectrum and receiving and transmitting packets, node 1's own packets are not displayed in the figure, as it does not overhear its own transmissions.

The figure presents how the general avoidance maneuver and network resynchronization take place using the principles of swarming:

Time $t = 1$ At the beginning of the experiment, the entire testbed network has settled on the 802.11 channel 1 (2.412 GHz) for

communication, while the signal generator is jamming channel 2 at 2.417 GHz. To begin the test, the interference signal is moved to the network's current communication frequency, channel 1.

Time $t = 2$ As soon as node 1 has detected the continuous presence of the interference source on its currently used channel 1 and decided that the signal is either completely blocking its access to the channel or too strong to maintain sufficiently good communication links to other nodes, it samples the spectrum noise floor for a suitable alternative channel.

Time $t = 3$ Node 1 decides that the adjacent channel 3 (2.422 GHz) provides a better communication environment, subsequently switches to this channel, and from now on sends out its IEEE 802.11 ad-hoc mode beacons on this new frequency.

Time $t = 4$ In the meantime, node 2 has noticed the absence of data packets and beacons from its neighbor node 1. After the switch of node 1 at time $t = 3$, it has now overheard several frames of node 1 on its new channel 3. Since it is not affected by the signal generator,² it is not aware of the degraded communication environment. However, due to the cohesion and alignment rules of the swarm algorithm, it also decides to do a frequency adjustment to channel 3.

Time $t = 5$ Once node 2 has switched its transmission frequency, it has reestablished the link to node 1 but lost its communication with node 3. Node 3 now notices that both nodes 1 and 2 are absent on the previously selected channel 1, but through a quick scan, identifies that these peers can now be found on the alternative frequency channel 3.

Time $t = 6$ After the decision of the two peers proves to be stable, node 3 also mirrors the decision made by its predecessors and follows to channel 3 according to the cohesion rule.

Through this iterative process based on local observation of the communication environment, the

¹The channels where no nodes communicated during this experiment are not displayed for the clarity of the presentation.

²Which is also the reason that, after time $t = 1$, both nodes 2 and 3 can continue to send packets on the now jammed channel 1 and these frames are decodable due to the lower header symbol rate.

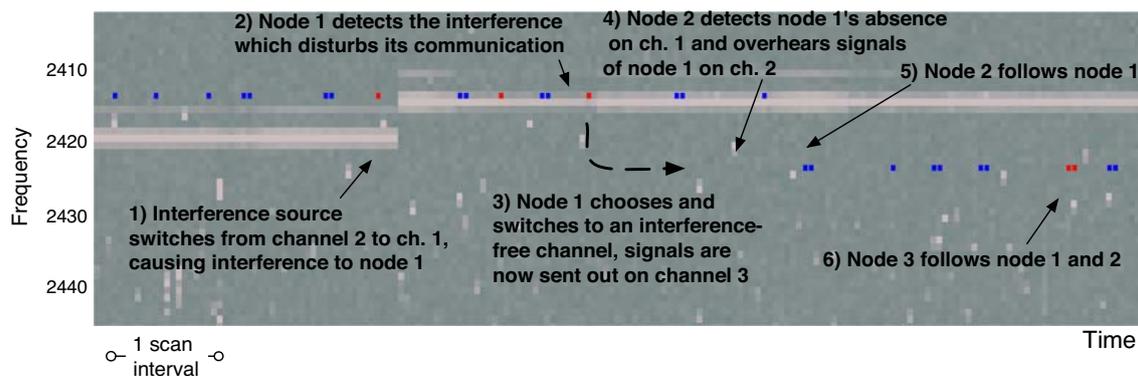


Fig. 9 Demonstration of environmental information is transmitted through the network through passive observation [11]

cognitive radio network has resynchronized on an alternative channel configuration. This behavior works analogously in more complex environments where nodes only have a partial view of the networks or the communication links are asymmetric: here, the same process is repeated in wave-like patterns throughout the network for all nodes in transmission range until the entire network has converged to the new frequency. This convergence will be the subject of further study in the following section, where we report on the configuration efficiency of the algorithm in larger environments, particularly those that (a) due to distance will require multihop routes through the network and (b) situations in which, due to spectrum heterogeneity, no common channel assignment may be found.

5.2 Network configuration performance

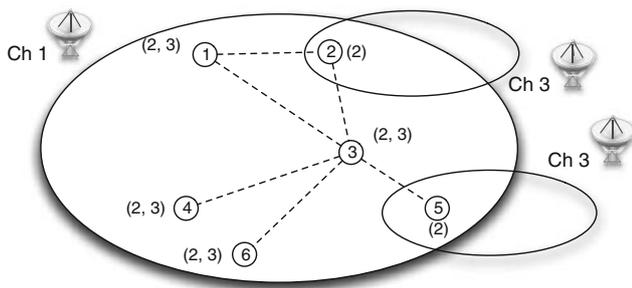
After the general concept of coordination through observation has been validated in a series of hardware experiments, we investigated the overall configuration performance of the local scheduling approach. In this part of the evaluation, we generated 1,000 concrete topologies and environmental conditions (node locations, local spectrum availability and links) for each of the four problem tasks that a cognitive radio network deployment would face in our reference scenario (see Section 4.1), and let a mathematical program using global knowledge determine the best configuration settings for each node in the cognitive radio network that would maximize the amount of feasible communication flows. The exact same topological setup was then also simulated (and for small instances also arranged in the hardware testbed) to determine the solution that the swarm algorithm would provide, through this process the configuration efficiency of the local control algorithm could be directly compared to the theoretical

optimum, thus allowing for a better comparability among different lines of research.

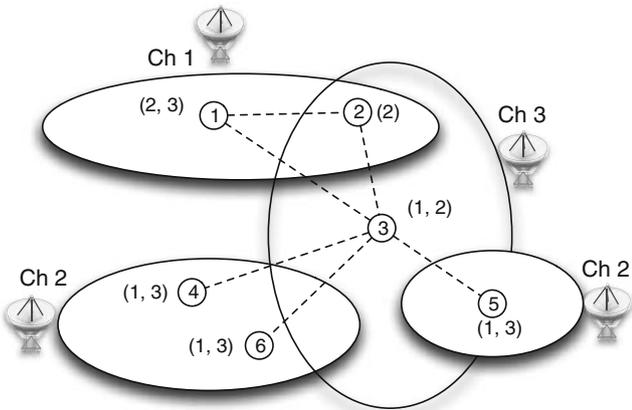
While this evaluation was performed for all four application tasks, due to space constraints, we will limit our discussion to two particular aspects from the four evaluation scenarios: (1) the configuration of a multihop cognitive radio network to a global, commonly available channel assignment that could be used in our reference scenario for the dynamic selection of a coordination channel among the home entertainment devices, and (2) the location-specific configuration to regionally available common channels that are linked together by bridging nodes that would perform spectrum-aware routing of messages. Such a setting will most likely occur in highly spectrum heterogeneous deployments or when high traffic demands among certain groups of devices require the allocation of more bandwidth and, consequently, more channels across the cognitive radio network. Figure 10 visualizes these two evaluation settings, which are specializations of the tasks “common channel assignment” and “spectrum-aware routing.”

This comparison between the swarm algorithm and the reference solution providing the best possible configuration can, however, be conducted in two different ways, (1) by contrasting the final configuration result, i.e., how much network communication flows can be fulfilled after the algorithms have finished their network coordination and configuration phase, and (2) by contrasting the steepness of the performance improvement, i.e., investigate the utility of intermediate configuration assignments³ and how large the incremental improvements in each iteration of the control

³While not being the final and most likely best outcome, intermediate configurations already allow some of the network nodes to communicate and, thus, already provide some utility to the cognitive radio network as a whole.



(a) Global Common Channel Assignment in a Multi-Hop Network



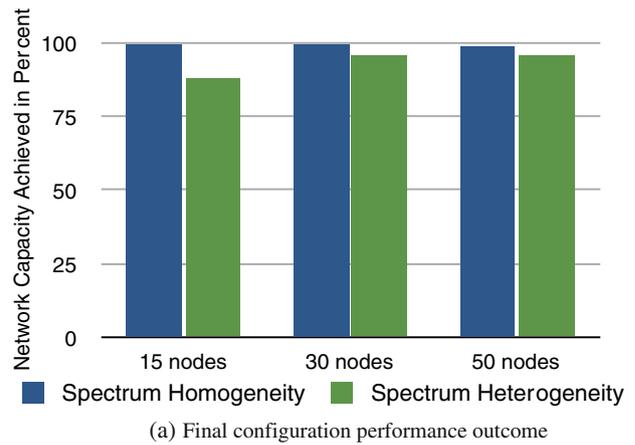
(b) Channel Assignment in Local Areas and Spectrum-Aware Routing

Fig. 10 When self-configuring in environments with homogeneous and heterogeneous spectrum availability, a cognitive radio network either needs to determine a globally workable common channel assignment or confined areas among which spectrum-aware routing needs to occur [9, 10]

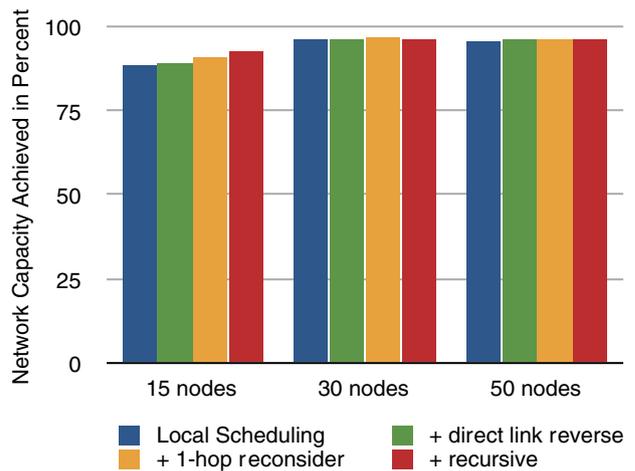
algorithms are. Especially this latter point will be important in dynamic environments where the cognitive radio network has to adapt to rapid and frequent changes, and it turns out to be a major advantage of the local control algorithm due to its decentralized and, thus, highly parallelized nature.

5.2.1 Final configuration efficiency

Figure 11a shows the final configuration outcome of the local control algorithm for network sizes of 15, 30, and 50 nodes for the two cases of multihop coordination under spectrum homogeneity and spectrum heterogeneity. As discussed before, the configuration efficiency is measured relative to the theoretical optimum, i.e., as the percentage of network communication flows that can be fulfilled when the network is configured by the local control algorithm compared to the flows that are theoretically feasible using the best configuration. As



(a) Final configuration performance outcome



(b) Improvements can be gained through additional local heuristics

Fig. 11 A comparison of configuration performance of the local control algorithm measured in percent of the theoretically possible optimum [9, 10] (a, b)

can be seen in the figure, the local control algorithm scores reasonably well, reaching a near perfect solution (95% and higher) for the case of spectrum homogeneity and a high configuration outcome of 88% (and higher) for the case of spectrum heterogeneity.

While the local control algorithm generally provides good performance, it is, however, remarkable that, for the more complex scenario, the configuration outcome seems to be slightly better in larger network sizes than in the smallest topology configurations, an observation that was consistently verified through more experiments and a wider array of network sizes. This issue was further investigated through manual inspection of the evaluated topologies and by following the configuration decisions that the local control algorithm would make in each setup, and led to the discovery of situations when a local control algorithm would only provide suboptimal outcomes.

The reason why the swarm algorithm performs worse in smaller scenario sizes can be attributed to the lower link-to-node ratio in these setups. In our experiments, we generated topologies with N nodes, which could, in a single-hop scenario, theoretically have up to $N \cdot (N - 1)$ feasible links among them. If these N nodes now generate traffic, there will overproportionally exist more “critical” bottleneck links in smaller networks than in larger deployments, where there might exist alternative routes between a source and a destination. This observation was validated through regression tests for which the model and the (only) statistically significant variable, network connectivity or average number of links that a node has, is shown in Table 2. Figure 12 visualizes this problem and a possible remedy in a simple example.

For the example, consider a small topology of five nodes, named nodes A–E, which have different local spectrum availability as indicated in parentheses at each node. The nodes are placed in a way that some traffic needs to be forwarded by intermediate nodes, and for illustration purposes, assume that each link may only be used in one direction.⁴ In Fig. 12a, both nodes B and E begin to source some traffic flow f_1 , f_2 to node C, and due to their local decision-making approach, they both decide to use the most direct route, their direct link to E, for this outgoing flow. If node C, however, also incurs a traffic demand later, say to node A, it will find that all links in its proximity are being used and cannot be utilized to fulfill its own demand. These situations occur as traffic flows might appear at any time and are not known a priori by a node, but more importantly due to the fact that each node incorporates only the flows it directly knows about (those it either sources, sinks, or forwards) into its decision making process. This is, however, an inherent property of any local decision making process, and presharing this knowledge will create other issues, as discussed in the earlier parts of this paper.

At this point, node C can either decide to reject the outgoing flow to the application layer or try to renegotiate with its neighbors to free up one of the blocked links. For this renegotiation, it could contact either node B or node E and inquire if either one of them could reroute its flow using an alternative route.

⁴If we assume the more realistic case of bidirectional links, the same bottleneck situations would occur due to the saturation of specific links, our assumption will, however, provoke these situations using fewer nodes and, therefore, only make the illustration less complex.

Table 2 Regression model: configuration efficiency and statistically significant parameter of average network connectivity

| Parameter | Coefficient | Error | p Value |
|----------------------|-------------|-------|-----------|
| (Intercept) | 91.62 | 0.74 | <0.001 |
| Network connectivity | 0.459 | 0.07 | <0.001 |

Such an inquiry could be realized in two different ways: (a) by introducing a special flag in the acknowledgement frames or by transmitting a special inquiry packet that could be read and interpreted by the remote control algorithm, or—as a transparent and, thus, more universal approach—(b) by exploiting certain response behaviors of the protocols using the connection, for example, temporarily stopping the flow by artificially introducing a TCP RST packet.

Independent of the concrete realization, node B may find after receiving the rerouting request that it can indeed redirect its flow to node C through the intermediate hop node E, as its links still have some remaining capacity, as shown in Fig. 12b. This consequently frees up the bottleneck link and node C can send its flow to destination A.

The situation becomes more complex when nodes engage in spectrum-aware routing of traffic as visualized in Fig. 12c, but it may still be addressed using the same simple mechanism. Here, the direct neighbors blocking a particular critical link might not be the original source of the flow but, rather, act as an intermediate node forwarding the packets as in the example of node B in Fig. 12c. In this case, the rerouting request could first be considered by the intermediate router node B or immediately passed on to the original source node D. Once either the source of a flow or the intermediate forwarding node discovers an alternative route, the bottleneck can be resolved as in Fig. 12d.

There might, however, exist more complex situations where the termination of individual flows will not provide significant benefit as a node due to high density of a deployment and many concurrent flows may not locally find any alternatives to send a flow. In these contexts, it could help to initiate a recursive inquiry for reroutes, i.e., if node C asks node B to reroute but B does not discover any other available route, it will ask its own neighbors in turn to consider a reroute as well. Figure 11b displays the improvement that can be gained from employing such heuristics locally at each node. The initial result of the “pure” algorithm can be boosted with each additional technique, so that even the nonrecursive, strict neighbor oriented algorithm will provide more than 93% of the theoretically feasible outcome. This performance gain cannot, however, be

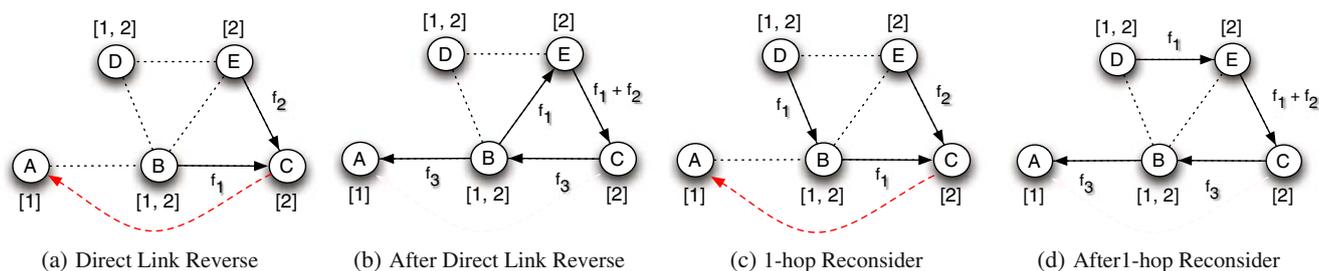


Fig. 12 Simple local heuristics can be used to free up critical bottleneck links and improve the configuration efficiency in special cases [10] (a–d)

achieved in scenarios of larger sizes, as the original issue of a low link-to-node density is not present there. It has to be noted, however, that even these additional heuristics will leave a small remaining margin of about 2–7% to the optimal solution. We found out that, in order to close this gap, the control algorithm needs to create sometimes highly complex and unintuitive routing decisions to remove and avoid cascading and interdependent bottlenecks. To achieve this level of fidelity, the control algorithm will need global knowledge of the network topology, state and flow demands, and sometimes extensive amounts of time to find this solution due to the nonpolynomial complexity of the required mathematical program. This high computation demand will make the computation of the reference solution for moderate or large network sizes expensive, if not—especially in very dynamic situations—infeasible.

5.2.2 Characterizing the improvement gradient

As the two cognitive radio control algorithms follow different operating strategies—the independent algorithm calculates the solution in parallel at each node while the centralized controller calculates and distributes the reference solution to all nodes—it is also important to analyze how the algorithms reach a configuration solution and quantify the intermediate steps and configuration states the algorithm will reach during the computation and distribution of the configuration assignment. This aspect of how fast an algorithm converges and how much utility already intermediate states will provide to the network will be the focus of the following analysis.

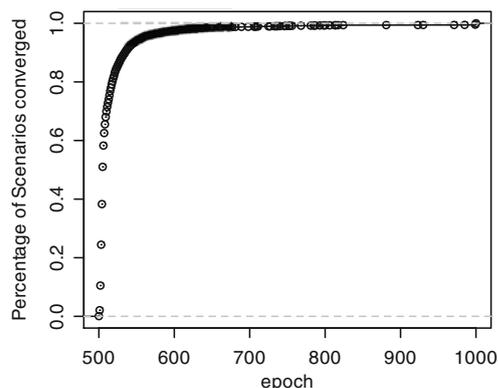
While the pure computation time of an algorithm, e.g., the time required to solve a mathematical program, is very relevant in practice and has proven to already be a significant factor for the moderate-size networks in our experiments, the following discussion will only focus on distribution aspects, i.e., we assume that each node and the central control authority can

instantaneously create a solution and study how long it will take to distribute the configuration instructions in the network, when the network will converge and how well partially distributed or partially locally computed solutions will configure the cognitive radio network.

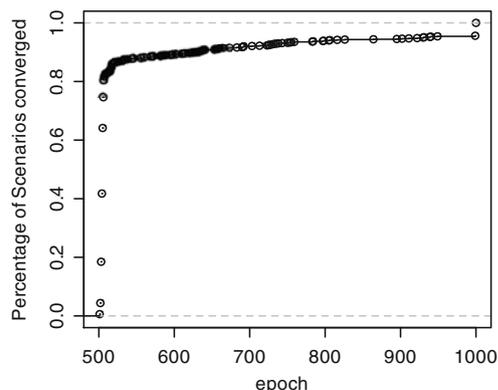
For this analysis, we extended our implementation on the hardware nodes and the software simulator by a second control algorithm. This algorithm was performing the tasks of a centralized spectrum server, coordinating the network nodes using global knowledge, which was collected using a dedicated control channel. While this control channel was not subject to any interference and was reserved for the use of the centralized controller only, it was limited in the number of coordination and configuration messages it could carry at any given point in time.

Figure 13 shows the convergence times for networks of sizes of 10 and 50 nodes, each controlled by the swarm algorithm implementation and the centralized controller. In these setups, five cognitive radio networks were deployed with varying node sizes and had to configure themselves in such a way that full network connectivity was established and the amount of interference was minimized for all network nodes. Each algorithm was given 500 epochs to build its data structures and for initial configuration. At epoch 500, an interference source was placed in the network such that 10% of all nodes were affected and unable to communicate on the current channel. At this time, the control algorithms had to adapt to this external event and configure the cognitive radio networks in such a way that complete communication in the network was reestablished.

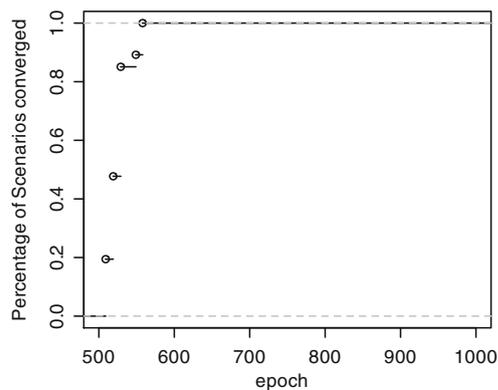
As can be seen in the figure, both algorithms coordinate the five networks of 10 nodes in a reasonably fast manner; after 30 iterations, full network coordination is achieved in more than 80% of the 10,000 simulation runs. As the network, however, grows to 50 nodes each, the centralized control scheme dramatically loses adaptation performance and reconfig-



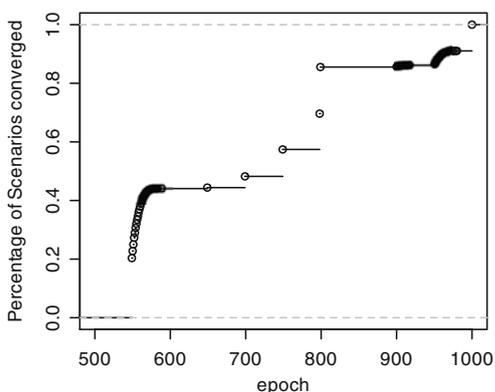
(a) Swarm Algorithm, 10 nodes



(b) Swarm Algorithm, 50 nodes



(c) Centralized Controller, 10 nodes



(d) Centralized Controller, 50 nodes

◀ **Fig. 13** Convergence performance for a dense deployment of five cognitive radio networks with varying sizes [13] (a, b)

uration agility, as the control channel required for all coordination messages becomes saturated and many nodes affected by this major reconfiguration event wait for instructions from the spectrum server: even after 300 iterations, more than 50% of the scenarios have not yet converged.

This highlights that it is not only important how *good* the configuration provided by the control algorithm actually is, but at least as important how *fast* the algorithm can distribute configuration instructions to the nodes. While the swarm algorithm would sometimes not achieve the full configuration potential (the remaining 2–7% as discussed in the previous section), the analysis of the configuration process shows that, especially in moderate and large network sizes, it achieves a final configuration state faster. Furthermore, the intermediate configuration states also provide sufficient configuration to allow, at parts of the network, communication from the beginning.

5.3 Interoperability with other cognitive radio algorithms

When placed in a later deployment, one cannot assume that there exists only one cognitive radio network and that all nodes are homogenous, i.e., all cognitive radios will be managed and coordinated using the exact same or a directly compatible control algorithm. In fact, it will be most likely the case that there exists a variety of secondary user networks that will be highly heterogeneous, the nodes are of a variety of different models and makes and are controlled by a broad selection of different, potentially incompatible algorithms.

It is therefore of great interest that any given control algorithm is able to coordinate with other networks and network nodes that are using a different and potentially incompatible algorithm. This incompatibility can arise from two different sources; first, the algorithm might use an alternative and potentially orthogonal decision-making approach that will make common synchronization actions difficult, or second, the algorithm might rely on the explicit exchange of coordination messages, which may not be understood by the other system.

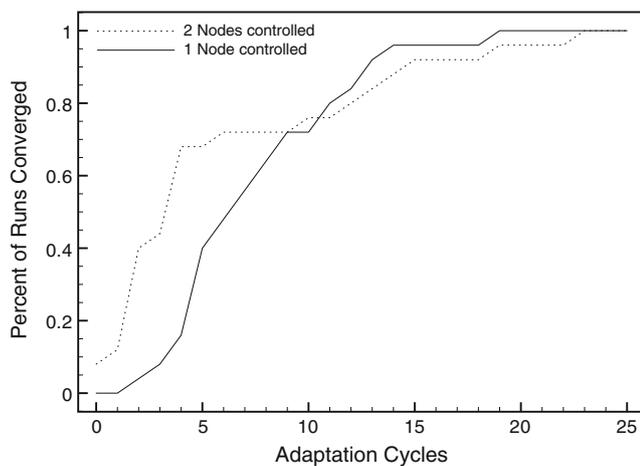
For this part of the evaluation, we analyzed whether cognitive radio network nodes driven by the swarm approach were able to coordinate with other nodes that were using a different protocol not understood by the swarm nodes. In these experiments, a number of nodes

in the hardware testbed were set up to be managed by a centralized control mechanism and were independently making adaptation decisions.

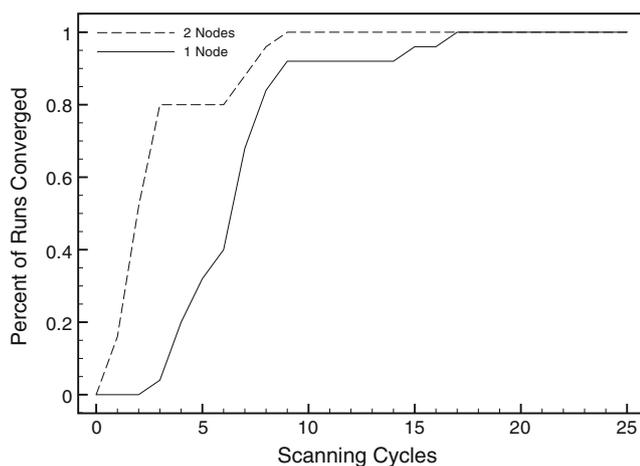
In the first part of these experiments, the remaining testbed nodes running the swarm algorithm were configured by the application layer to view the other network nodes as being “friendly” and to always maintain a cohesive network with them. When these nodes were performing adaptation decisions, the nodes controlled by the swarm algorithm could only remotely observe their adaptation actions (as they could not understand the other algorithm’s reconfiguration instructions) and adapt themselves to recreate a working network. Figure 14a shows the cumulative adaptation performance of a heterogeneous cognitive radio network where some nodes were controlled by an

incompatible algorithm performing independent adaptation decisions, and network nodes controlled by the swarm algorithm had to observe and adapt in such a way that network connectivity was maintained. As can be seen in the graph, network connectivity is quickly reestablished, in 80% of the experimental trials, complete network coordination is regained after 10 scanning cycles. The network coordination is even faster as more nodes are externally controlled by the incompatible algorithm as more observations about the other stations’ adaptation decisions can be made by the swarm nodes and faster reconfiguration decisions are being made.

In the second part of these experiments, the swarm nodes were instructed by the application layer to consider the incompatible network nodes as being “unfriendly” and that their interference-generating presence should be avoided whenever possible. In this experiment, these externally controlled nodes were repeatedly selecting the channel the swarm nodes had chosen for communication, after which the nodes controlled by swarm behavior were each independently trying to vacate the channel and rendezvous on a new communication frequency. Figure 14b shows the cumulative adaptation performance when the cognitive radio network had to vacate and resynchronize after such take-over events. As can be seen in the figure, in 80% of the experimental runs, the network nodes have independently selected and rendezvoused on an alternative frequency in 10 scanning cycles or less. Again, if more nodes are externally controlled, more observations are present for the swarm algorithm and adaptation decisions can be made more rapidly.



(a) Coordination with incompatible algorithm.



(b) Avoidance of incompatible algorithm.

Fig. 14 Convergence of swarm algorithm when exposed in a heterogeneous network environment where other cognitive radio networks with incompatible control algorithms are present [11] (a, b)

5.4 Dependency on node-activity levels

Besides showing robustness to and interoperability with other, potentially incompatible configuration schemes by external nodes, it is also important that the algorithm itself is robust to varying parameters and behavior of the nodes themselves. One of the properties that could vary largely among deployment scenarios is the activity level of the nodes, i.e., the amount of time a node is awake, ready to receive, listening to incoming packets, or transmitting itself. An extreme case of this particular parameter is, for example, sensor networks where the nodes are actually dormant most of the time and only wake up in certain intervals to communicate, yet also in a wireless network as part of a home entertainment system, many nodes will typically not be actively engaged in wireless communications (listening, receiving, or sending) at all times. This section therefore investigates whether the control algorithm is able

to coordinate networks at various node-activity levels and if this parameter has a significant impact on the overall configuration efficiency.

Whenever nodes are not awake all the time, the node sleep cycles create a problem for network management, as during the times a node is asleep it cannot engage in any data communication and will, thus, not participate in configuration and coordination efforts. To study the impact of the node activity levels on configuration performance, we again created networks for various topologies and network sizes and let each network node be dormant at each point in time with a certain probability. Whenever a cognitive radio is in sleep mode, it is not able to receive or send any messages and will therefore neither be able to receive any other network traffic that might be used for coordination, nor can it indirectly announce its presence by engaging in any type of data communication.

Figure 15 shows the average convergence time for a deployment of five networks with five to 70 nodes each. Each of the network nodes independently determines whether it will currently go into sleep mode or be awake; the figure shows the average time it takes all five networks to select an appropriate transmission frequency such that all nodes of a given network can communicate while minimizing interference at node activity levels between 25% and 100%. As can be seen from the graph, the control algorithm is able to coordinate the cognitive radio network independent of node activity levels, i.e., for the convergence times it is not relevant how many network nodes are currently participating in any type of communication.

While this outcome seems counterintuitive at first, there still exists a logical and simple explanation for it. If a significant proportion of a given cognitive radio network is at sleep, these nodes do not actively participate in the network, and due to the absence of any outgoing

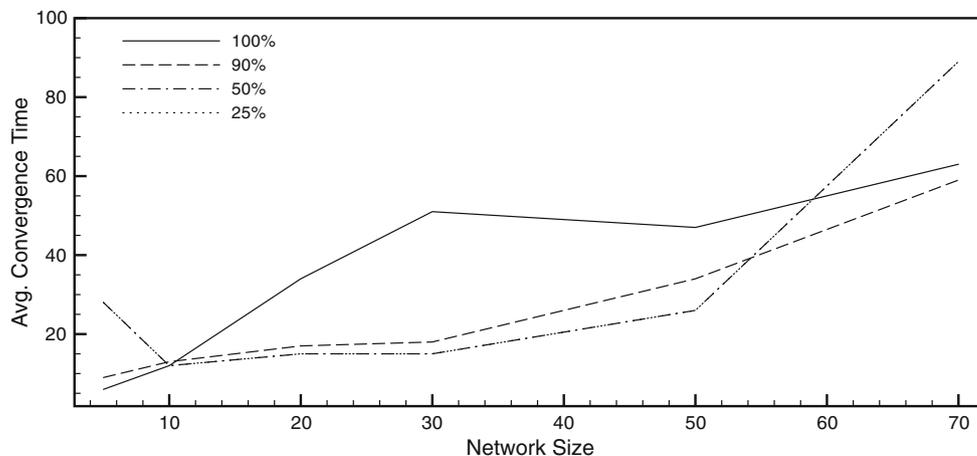
packets from them (data, ARP requests, beacons, etc.), the other network nodes are unaware of their existence and their current configuration. The resulting subset of the network, the part that is actively participating in the configuration synchronization, is, therefore, significantly smaller and consequently able to come to a common configuration consensus much faster: parts of the “long convergence tail” (see Fig. 13b) are cut off. As the active part of the population fluctuates over time, some correctly configured nodes will enter sleep mode while some not yet configured ones will awaken. These recently awoken nodes now discover a cluster of peers that have already converged to a common configuration and can immediately join them. This balancing behavior will continue even as the network reaches the extreme case of bimodal oscillation, where the same subset of nodes is always awake at the same time—resulting in complete coordination among the two clusters separated in time.

5.5 Convergence as a function of the amount of spectrum

Finally, the last part of the evaluation investigates how well our proposed local control algorithm scales with the amount of resources it is supposed to manage, i.e., the amount of spectrum it needs to monitor to rendezvous with other network nodes. While current 802.11 hardware is limited to 11 channels, future adaptive networking systems that are engaging in DSA over others parts of the spectrum will have access to dozens, if not hundreds, of orthogonal channels. This section investigates the impact on convergence time when each cognitive radio will have a large number of resources at hand.

For this study, we repeated our previous experiments but modified the software simulator to offer a

Fig. 15 Convergence time as a function of network size and node activity levels



larger number of channels to the cognitive radios in the network. Figure 16 shows the algorithm’s achieved percentage of theoretically feasible capacity over time for network sizes of 15, 30, and 50 nodes for a variety of spectrum sizes between 11 and 100 channels. In this

series of experiments, the nodes were configured to possess a single wireless interface only and be able to scan and decode packets up to three channels from their current transmission frequency, thus, not able to perceive large parts of the spectrum. As can be seen in the figure, if the amount of spectrum that the cognitive radio can access and, therefore, needs to listen to, increases, the networks generally achieve a lower and more slowly growing system capacity. This is true across varying network sizes, but in a larger network, this effect is more profound. This phenomenon of more slow and bounded capacity was also observed in the other simulations previously discussed.

The scanning width of three channels only was specifically chosen to mimic the most likely limited abilities of a low-cost cognitive radio system. There exist multiple ways to enhance the convergence of the algorithm in such situations, for example, through the additional rules as previously discussed or through technical means, as, for example, introducing additional sensing capabilities to the cognitive radio. In order to enable the node to scan and detect stations on a larger number of channels, the cognitive radio could either be equipped with additional wireless interfaces dedicated to scanning purposes, a wide-band spectrum analyzer, or some other means to actively probe the available channels.

In order to test to what extent the swarm-behavior-based control algorithm could make use of and be enhanced by such devices, we equipped the cognitive radio nodes with a WiSpy (v1) device, a low-cost (<100 US-\$) energy detector for the entire 802.11 spectrum. This device can sense energy levels across the frequency spectrum within a few milliseconds and is, thus, orders of magnitude faster than a scan-and-decode sweep performed by the wireless interface. The disadvantage of using such a low-cost energy sensor is its simplicity: if the scanner detects a high amount of energy in a certain portion of the band, it can still not be sure whether this energy is emitted from its cognitive radio network peers or by other interfering stations, and thus, the situation still requires further inspection. Nevertheless, such additional information might be helpful in zooming into those portions of the available channel spectrum that show some significant activity and, therefore, help the cognitive radio save resources during its learning phase.

There exist multiple policies on how this information gained by the energy sensor might be used to inform the detection process. One of the simplest possible strategies could be to use the energy measurements to locate the most active parts of the spectrum and

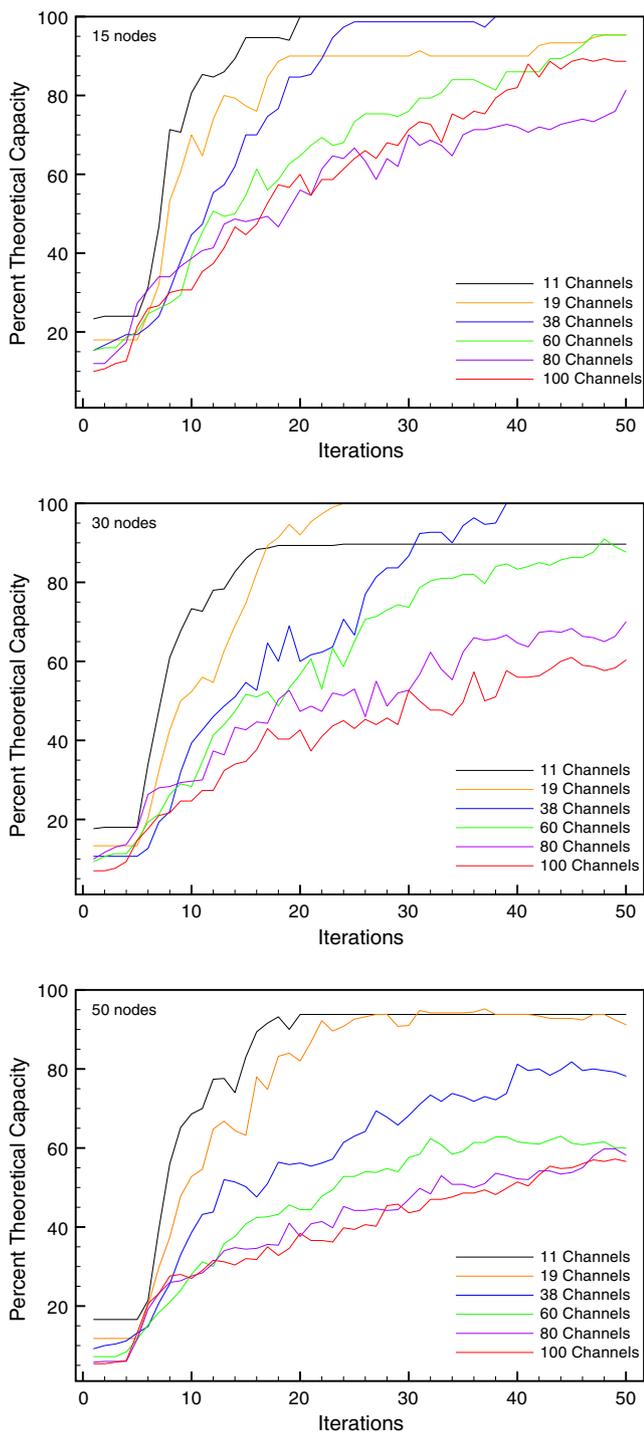


Fig. 16 Convergence of the swarm algorithm depending on the number of totally available channels

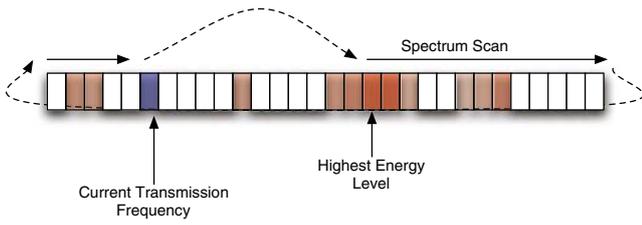


Fig. 17 Example scanning routine to skip unpopulated parts of the spectrum

begin the channel search there, thus avoiding the necessity to sweep over unoccupied parts of the spectrum. Figure 17 shows how such a strategy would be used by the cognitive radio. As can be seen easily, even such simplistic use of the energy sensor would provide an $\frac{n}{2}$ improvement over the amount of channels being scanned during a routine sweep (as introduced by the additional rules as previously discussed), which was also confirmed through simulations.

Figure 18 shows the effect on the algorithm’s convergence and overall achieved system capacity when augmented through this simple energy-sensing strategy for 50 nodes and varying spectrum sizes, where the solid line indicates the performance of that particular configuration described in the legend and the dashed line of the same color indicates the performance when additionally using the energy sensor. As can be seen in the graph, the system’s overall convergence and capacity is significantly improved through the use of the energy sensor, even when using this highly simple decision strategy, and with its assistance, the originally poor performance in large channel sets can be overcome.

A second, somewhat more sophisticated search strategy could be to use the energy measurements to guide

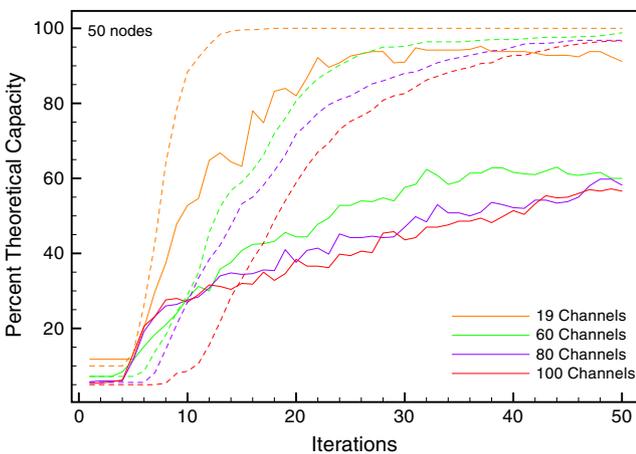


Fig. 18 Improvements on convergence time obtained by energy sensing strategy 1

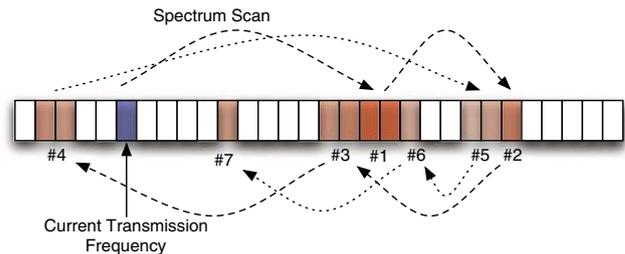


Fig. 19 Example scanning routine based on ranked energy levels

a random access search across the channels. After the energy sensor has reported which frequencies currently show the highest energy levels and, therefore, potentially the most stations, the cognitive radio could consecutively monitor only those preselected frequencies in a descending order to scan for peers. Figure 19 visualizes this search strategy, and Fig. 20 quantifies the improvements in an analogous way as Fig. 18. As shown in the figure, the slightly more sophisticated search strategy also provides improvements in terms of convergence time and achieved capacity, even more than the first strategy.

The fact that the swarm intelligence-based cognitive radio control algorithm may synchronize a network even across a large amount of channels with only a limited scanning window within a reasonable amount of time (an iteration of the algorithm could be set close to the beaconing interval of 104 ms) makes it, together with its low computational complexity and robustness to node activity levels, a viable candidate in many deployment scenarios. It is interesting to note that the algorithm is able to incorporate information provided by an additional external sensor and makes efficient use

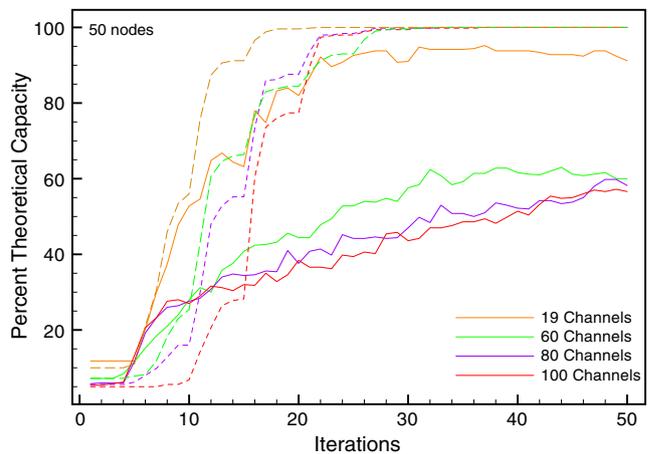


Fig. 20 Improvements on convergence time obtained by energy sensing strategy 2

of it, even if the data only provides a limited view and/or resolution.

6 Conclusions

This paper discussed the concept of local control of cognitive radio networks through the principles of swarm behavior as found in schools of fish and flocks of birds. We presented an extended discussion of why the concept of swarm intelligence is a good approach for the efficient coordination of cognitive radio networks, discussed the adaptation process necessary to use such biological algorithms in technical problem sets, and provided an extended example of how this algorithm can address the problems of cognitive radio control.

Beyond optimizing its internal operation and reconfiguration procedures, it is necessary for a cognitive radio to incorporate information about neighboring wireless network nodes and other sources of interference. As the number of nodes increases, it is difficult and costly to obtain sufficient information for making appropriate frequency-usage decisions, particularly based on approaches that share information globally or that require centralized control. Due to the difficulties in obtaining global knowledge about spectrum usage across a radio network, it is worthwhile to explore whether adequate configuration decisions can be determined when there is no central authority to determine spectrum usage or the network nodes do not possess knowledge of the entire network topology. We propose a method that relies on information that can be locally observed by each node. If it is possible to obtain sufficient network coordination by basing the decision on local knowledge only, such a cognitive radio system would enjoy higher efficiency, better scalability, and increased robustness, as the bottleneck of explicit coordination communication is removed.

To exploit the value of local information, we propose the use of control models based on the principles of swarm intelligence. The collective and seemingly complex behavior of “swarming” as expressed by schools of fish and flocks of birds is actually the result of a set of simple rules that are followed by each individual in the group. Yet, when each member follows these rules, everyone’s local behavior and the resulting interactions between individuals give rise to a complex global behavior exhibiting properties beyond any behavioral response encoded in the original rule set. In this paper, we distill and translate the essential aspects of swarm intelligence to the problem of dynamic spectrum usage. We show that a rather straightforward set of requirements can be adapted from swarming behavior

to provide proper configuration decisions to small and large sets of cognitive radio networks operating in a variety of situations.

To examine the applicability of local control to cognitive radio networking, we first examined the requirements and challenges of dynamic spectrum usage. We next considered the problem of distributed cognitive radio coordination based on the principles of swarm intelligence. We demonstrated how the principles of repulsion and attraction (concepts central to swarm intelligence) can be applied to cognitive radio coordination by presenting the results of simulation based on these principles. To confirm our simulation results, we developed a working prototype of a cognitive radio network controlled by the principles of swarm behavior and demonstrated the feasibility of local control in a hardware testbed using off-the-shelf commodity network nodes. We find that this type of control algorithm is well suited to efficiently configure wireless networks based on local information only, thus making the need for expensively collected global environmental information unnecessary and allowing for effective coordination even of large networks.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. e2r (2009) FP6 end-to-end reconfigurability (ER) integrated project (IP). <http://www.e2r.motlabs.com>
2. MadWifi (2009) The MadWifi project. <http://www.madwifi.com>
3. WiSpy (2009) WiSpy spectrum analyzer. <http://www.metageek.net>
4. Camazine S, Deneubourg J-L, Franks NR, Sneyd J, Theraula G (2001) Bonabeau E Self-organization in biological systems. Princeton University Press, Princeton
5. Cao L, Zheng H (2005) Distributed spectrum allocation via local bargaining. In: SECON
6. Clark CW, Ducas R (1994) Balancing foraging and antipredator demands: an advantage of society. *Am Nat* 144:542–548
7. Couzin ID, Franks NR (2003) Self-organized lane formation and optimized traffic flow in army ants. *Proc R Soc Lond B* 270:139–146
8. Doerr C, Colagrosso M, Grunwald D, Sicker DC (2008) Scalability of cognitive radio control algorithms. In: Proceedings of the IEEE international symposium on wireless pervasive computing (ISWPC)
9. Doerr C, Grunwald D, Sicker DC (2008) Dynamic control channel assignment in cognitive radio networks using swarm intelligence. In: Proceedings of the IEEE global communication conference (GLOBECOM)

10. Doerr C, Grunwald D, Sicker DC (2008) Dynamic control channel management in presence of spectrum heterogeneity. In: Proceedings of the IEEE military communication conferences (MILCOM)
11. Doerr C, Grunwald D, Sicker DC (2008) Local independent control of cognitive radio networks. In: Proceedings of the international conference on cognitive radio oriented wireless networks and communications
12. Doerr C, Sicker DC, Grunwald D (2007) Optimizing for sparse training of cognitive radio networks. In: Proceedings of the first international workshop on cognitive wireless networks (CWNets)
13. Doerr C, Sicker DC, Grunwald D (2007) What a cognitive radio network could learn from a school of fish. In: 3rd international wireless internet conference
14. Hamner WM (1984) Aspects of schooling in euphausia superba. *J Crustac Biol* 4(1):67–74
15. Ileri O, Samardzija D, Mandayam NB (2005) Demand responsive pricing and competitive spectrum allocation via a spectrum server. In: First IEEE international symposium on new frontiers in dynamic spectrum access networks
16. Kelly K (1994). *Out of control: the new biology of machines, social systems and the economic world*. Addison-Wesley, Reading
17. Kitano H (2004) Biological robustness. *Nature* 5:826–837
18. Kurkowski S, Camp T, Colagrosso M (2005) Manet simulation studies: the incredibles. *SIGMOBILE Mob Comput Commun Rev* 9(4):50–61
19. Kurkowski S, Camp T, Navidi W (2006) Two standards for rigorous manet routing protocol evaluation. In: 2006 IEEE international conference on mobile adhoc and sensor systems (MASS)
20. Lee WC (1991) *Mobile design fundamentals*. Wiley, New York
21. Neel JO, Reed JH (2006) Performance of distributed dynamic frequency selection schemes for interference reducing networks. In: Military communications conference MILCOM
22. O'Brien DP (1989) Analysis of the internal arrangement of individuals within crustacean aggregations (euphausiacea, mysidae). *J Exp Mar Biol Ecol* 128:1–30
23. Raman C, Yates RD, Mandayam NB (2005) Scheduling variable rate links via a spectrum server. In: First IEEE international symposium on new frontiers in dynamic spectrum access networks
24. Reynolds C (1987) Flocks, herds, and schools: a distributed behavioural model. *Comput graph* 21:25–34
25. Rondeau TW, Le B, Rieser CJ, Bostian CW (2004) Cognitive radios with genetic algorithms: intelligent control of software defined radios. In: SDR forum technical conference
26. Vogel S (2005) *Cats' paws and catapults: mechanical worlds of nature and people*. W. W. Norton, New York
27. Weingart T, Yee GV, Sicker D, Grunwald D (2006) A dynamic cognitive radio configuration algorithm. *IEEE Commun*
28. Zheng H, Cao L (2005) Device-centric spectrum management. In: First IEEE international symposium on new frontiers in dynamic spectrum access networks