This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

A Review of Computational Intelligence Techniques in Wireless Sensor and Actuator Networks

Nicolas Primeau*, Rafael Falcon†*, *Senior Member, IEEE*,
Rami Abielmona†*, *Senior Member, IEEE*, Emil Petriu*, *Life Fellow, IEEE*
* School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
† Research & Engineering Division, Larus Technologies Corporation, Ottawa, Canada
Email: nprim050@uottawa.ca, rafael.falcon@larus.com, rami.abielmona@larus.com, petriu@uottawa.ca

Abstract*—Wireless sensor and actuator networks** **(WSANs) are heterogeneous networks composed of many different nodes that can cooperatively sense the environment, determine an appropriate action to take, then change the environment's state after acting on it. As a natural extension of Wireless Sensor Networks (WSNs), WSANs inherit from them a variety of research challenges and bring forth many new ones. These challenges are related to dealing with imprecise and vague information, solving complicated optimization problems or collecting and processing data from multiple sources.** ***Computational Intelligence **(CI) is an overarching term denoting a conglomerate of biologically and linguistically inspired techniques that provide robust solutions to NP-hard problems, reason in imprecise terms and yield high-quality yet computationally tractable approximate solutions to real-world problems. Many researchers have consequently turned to CI in hope of finding answers to a plethora of WSAN-related challenges. This paper reviews the application of several methodologies under the CI umbrella to the WSAN field. We describe and categorize existing works leaning on** ***fuzzy systems*****,** ***neural networks*****,** ***evolutionary computation*****,** ***swarm intelligence*****,** ***learning systems*** **and their** ***hybridizations*** **to well-known or emerging WSAN problems along five major axes:** ***actuation*****,** ***communication*****,** ***sink mobility*****,** ***topology control*** **and** ***localization*****. The survey offers informative discussions to help reason through all the studies under consideration. Finally, we point to future research avenues by (a) suggesting suitable CI techniques to specific problems, (b) borrowing concepts from WSNs that have yet to be applied to WSANs or (c) describing the shortcomings of current methods in order to spark interest on the development of more refined models.**

***Index Terms*—wireless sensor and actuator networks, computational intelligence, fuzzy systems, neural networks, evolutionary computation, localization, sink mobility, topology control**

LIST OF ABBREVIATIONS

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
AES	Artificial Endocrine System
AFSA	Artificial Fish School Algorithm
AIS	Artificial Immune System
ANN	Artificial Neural Network
AOI	Area of Interest
AOA	Angle of Arrival
ART	Adaptive Resonance Theory
BA	Bees Algorithm
BBA	Biogeography-based Optimization Algorithm
BFA	Bacterial Foraging Algorithm
CHNN	Competitive Hopfield Neural Network
CI	Computational Intelligence
CRNDP	Constrained Relay Node Deployment Problem
CS	Cuckoo Search
DE	Differential Evolution
DL	Deep Learning
DSS	Decision Support System
EA	Evolutionary Algorithm
FA	Firefly Algorithm
FIS	Fuzzy Inference System
FL	Fuzzy Logic
GA	Genetic Algorithm
GPS	Global Positioning System
GSO	Glowworm Swarm Optimization
HS	Hybrid System
HaS	Harmony Search
IEEE	Institute of Electrical and Electronic Engineers
LS	Learning System
MDP	Markov Decision Process
MLE	Maximum Likelihood Estimation
MOEA	Multi-Objective Evolutionary Algorithm
MOO	Multi-Objective Optimization
MOPSO	MultiObjective Particle Swarm Optimization
MOVNS	MultiObjective Variable Neighbourhood Search
MRTA	Multi-Robot Task Allocation
NP	Non-Polynomial
NSGA-II	Non-Dominated Sorting Genetic Algorithm II
PC-TSP	Prize-Collecting Traveling Salesman Problem
PSO	Particle Swarm Optimization
QoS	Quality of Service
RL	Reinforcement Learning
RSN	Robotic Sensor Network
RSSI	Received Signal Strength Indication
SCX	Sequential Constructive Crossover
SI	Swarm Intelligence
SIA	Swarm Intelligence Algorithm
SOM	Self-Organizing Map
SVM	Support Vector Machine
TOA	Time of Arrival
TSP	Traveling Salesman Problem
TSP-N	Traveling Salesman Problem with Neighborhoods
TSP-TW	Traveling Salesman Problem with Time Windows
UAV	Unmanned Aerial Vehicle

WRSN	Wireless Rechargeable Sensor Network
WSAN	Wireless Sensor and Actuator Network
WSARN	Wireless Sensor, Actuator and Robot Network
WSN	Wireless Sensor Network

I. Introduction

Wireless sensor and actuator networks (WSANs) are a natural extension of wireless sensor networks (WSNs) [1] [2]. While WSNs usually consist of numerous, often weak computational and low-energy nodes, WSANs integrate a small number of resource-rich sinks and actuators into their topology. These sinks and actuators are able to change the state of the environment with their actions, hence closing the control loop on the underlying monitoring system. This allows for greater potential for computations in the network.

WSANs have witnessed many successful applications, from agricultural [3] to industrial cyber-physical systems [4], critical infrastructure protection [5] [6], smart homes [7] and autonomous animal control, such as the bull breeding paddock example reported in [8]. However, a growing potential for research and development is still largely untapped. While today's WSANs are limited in their actuation, reasoning, and sensing abilities, the WSANs of the future will likely have even more heterogeneous nodes and be deployed on a massive scale.

In order to achieve this, several challenges must be tackled and resolved. Many of the optimization problems found in WSANs are quite difficult to solve and their optimal solution cannot be identified within a reasonable amount of time. To make matters worse, WSANs have inherited the problems of WSNs, thus making communication unreliable. Additionally, WSANs exhibit a more complex and heterogeneous topology with different types of nodes. Overcoming these challenges requires a departure from classical problem-solving methods.

Computational Intelligence (CI) is another vibrant research field [9] [10] that encompasses a broad range of intelligent techniques such as fuzzy systems, neural networks, evolutionary computation, swarm intelligence, granular computing and other learning and optimization paradigms. The common denominator behind all the methodologies under the CI umbrella is their ability to process imprecise information and seek approximate yet good-enough solutions to these problems while ensuring their robustness and computational tractability. Another distinctive feature behind CI techniques is that they are often inspired by biological processes. For instance, evolutionary algorithms mimic natural evolution to converge to near-optimal solutions in an optimization problem, and artificial neural networks borrow inspiration from the brain's massively parallel architecture for processing their input data. All these CI techniques are suitable for WSANs given their robustness and tolerance for imprecision; therefore, many researchers have started applying them with great promise.

This survey has three main objectives. First, it will introduce key concepts such as CI, WSAN, and relevant WSAN problem definitions along five major axes: *actuation*, *communication*, *sink mobility*, *topology control* and *localization*. Then, an extensive list of CI techniques applied to these WSAN problems is unveiled and discussed to provide an overview of the status quo in this exciting research area. A final important objective will be to identify future trends and research opportunities to better guide subsequent research endeavors.

The works discussed here have been gathered from relevant journals, workshops, and conference proceedings. Some of them are thesis works. The survey does not aim to give an exhaustive view of the field, as we found that the pace of the emerging developments is quite astonishing. Instead, we seek to outline representative studies offering different perspectives on WSAN problems tackled by CI methods. Most of the referenced works are very recent, which helps visualize the directions and trends current research efforts are pursuing.

The rest of the paper is organized as follows: we justify the need for this review article in Section II by comparing it with several other relevant survey papers. Core definitions pertaining to the CI and WSAN worlds are respectively introduced in Sections III and IV to contextualize the rest of the survey. Then, the problems plaguing WSANs are unveiled in Section V. Section VI elaborates on the surveyed works and categorizes them along five major axes corresponding to the WSAN problems they attempt to solve. Section VII offers informative discussions on the problems and works reported, again categorized per WSAN problem. Similarly, Section VIII sheds light on some future trends and opportunities under each problem category while Section IX formulates some concluding statements.

II. Related Survey Papers

The purpose of this survey is to provide a broad overview of the application of CI techniques to problems that are inherently found in WSANs. Although both CI and WSANs are quite vibrant research fields, they are rarely mentioned together in a holistic manner. To the best of our knowledge, this is the first time that CI applications to the WSAN realm are systematically dissected, categorized and put together in a review article, hence bridging the gap between these two seemingly disconnected yet highly complementary paradigms. There exist, however, several published works that cover in depth multiple niche areas found in our survey. This section will provide an overview of some of those relevant studies.

Survey papers or books devoted to WSANs [11] [12] [13] [14] [15] are relatively scarce compared to those dedicated to WSNs as a whole [16] [17] [18] or to a particular WSN problem [19] [20] [21] [22] [23] [24]. In these studies, CI is not the focus of attention even if some of the solutions discussed therein hinge on any method under the CI canopy.

Some researchers have published valuable studies highlighting the utilization of several CI methods in a particular context/problem pertaining to WSNs [25] [26]. Others have reported on the applications of a certain CI technique to numerous problems revolving around WSNs [27] [28] [29] [30] [31] [32] [33] [34]. While useful in constructing the present survey, these works have a narrower scope than ours and are not geared towards WSANs.

In 2011, Kulkarni et. al. surveyed CI applications to the WSN field [35]. This well-cited paper gave an excellent overview of key WSN problems such as design and deployment, localization, routing and clustering, among others. It also breaks down the family of CI techniques into neural networks, fuzzy logic, evolutionary algorithms, swarm intelligence, artificial immune systems, and reinforcement learning. The influence of this survey is visible throughout our article. However, WSANs pose a unique set of challenges (such as actuation, sink mobility or topology control via mobile actors) and call for the redefinition of typical WSN problems (such as routing and clustering). CI applications to these novel and challenging scenarios were beyond the scope of the survey in [35] and hence left unhandled. Additionally, the CI field is being constantly reshaped by numerous contributions and our survey aims to present a fresher look at the categorization of the CI techniques [10].

Along the same lines, Abraham et. al. [36] published a Springer volume on CI applications to WSNs in 2017. This book is not as comprehensive as the survey in [35] but offers a variety of solution approaches to important WSN problems such as attack detection, cost-sensitive control, traffic state estimation and information security.

In summary, this survey attempts to give a top-down view of the state-of-the-art regarding CI usage in WSANs. The review is centered around a plethora of problems brought forth by WSANs instead of WSNs, how CI techniques have succeeded in tackling them, and provides a more recent view of the CI discipline. Our paper is an ambitious effort to capture the interplay between CI and WSANs, instead of delving into one particular CI technique or one WSAN problem exclusively. The motivation behind this survey is to provide both CI and WSANs researchers a glance of the intersection between these two fields at a higher level.

III. Computational Intelligence Techniques

The definition of CI is still not well agreed upon but some concepts are understood to be fundamental to the discipline. The methods are nearly always heuristic in nature, meaning they provide solutions that may be inexact, not always optimal, or imprecise. CI algorithms are often, if not always, inspired by nature, drawing some ideas but not precisely replicating the mechanism seen in nature. The Institute of Electrical and Electronics Engineers (IEEE) Computational Intelligence Society defines CI in their constitution, Article I, Section 5 as "*the theory, design, application, and development of biologically and linguistically motivated computational paradigms emphasizing neural networks, connectionist systems, genetic algorithms, evolutionary programming, fuzzy systems, and hybrid intelligent systems in which these paradigms are contained*" [10].

This definition will be the default guideline of this survey. The authors in [10] elaborate on six CI categories, but they consider three of them to be foundational, viz evolutionary computation, artificial neural networks and fuzzy logic. The following survey will consider five categories: *Evolutionary Algorithms* (EAs), *Swarm Intelligence* (SI), *Fuzzy Logic* (FL), *Learning Systems* (LS), and a fifth category for *Hybrid Systems* (HS). Table I outlines several important characteristics of these techniques in the CI family. The rest of this section elaborates on these techniques and showcases their applications to the five major categories of WSAN problems described in this survey. Figure 1 illustrates all CI methods found in the reviewed papers at a glance.

A. Evolutionary Algorithms

Evolutionary algorithms attempt to mimic natural evolution to discover appropriate solutions to an optimization problem [42]. A typical EA maintains multiple individuals in its population, where each individual (a.k.a chromosome) is a series of genes. This population changes over time, with operators working at the individual or group levels to create new individuals or change the existing ones. One iteration of such an algorithm usually performs the following actions: first, the population is modified by the genetic operators, typically mutation and crossover; the former alters existing chromosomes into new ones and the latter generates new chromosomes from at least two existing ones. Then, the individuals in the current population are evaluated in terms of their fitness function(s), with only the fittest individuals surviving to form the population in the next iteration. The fitness function could be either maximized or minimized depending on the problem at hand. This leads to a population of high-quality (elite) solutions. The previous steps are repeated until a predefined stopping criterion is met, e.g., reaching a threshold in the maximum number of iterations, fitness function value or execution time. The best individual in the population as defined by the fitness function is then adopted as the solution for the optimization problem. Other aspects often taken into consideration are diversity preservation in the population and what to do with infeasible individuals that emerge throughout the search process.

From a mathematical perspective, this is analogous to searching for the best combination of values that will optimize the fitness function, making them apt for combinatorial and numerical optimization problems. In consequence, it is possible and often likely that these fitness values drive the algorithm towards a local, and not global, optimum. These solutions may only escape such local optima through the genetic operators that often include problem-specific knowledge. While there is an expectation that the quality of the discovered solutions will improve over time, there is no guarantee whatsoever in terms of the optimality of the final solution.

This definition is vague and not all-encompassing. There can be more components to this, such as constraints on individuals, making them not unable to be part of the population, or modifications of the steps in the iteration. The following will quickly review the EA applications that have been uncovered in this survey.

1) Genetic Algorithms: (GAs) are one of the first and most prominent EA manifestations. Though they are not confined to optimization problems [43], they are usually used in this context. A typical GA will follow the iterative process described above, usually starting with a randomly generated population. The mutation operator slightly modifies existing solutions whereas the crossover operator generates offspring solutions that may be part of the next generation. There are many variants of these operators that are tailored to produce

TABLE I
OVERVIEW OF DIFFERENT TYPES OF CI TECHNIQUES

Type of CI Technique	Computational Complexity	Application Scenarios	Example
Fuzzy Logic	Low	Reasoning with vague and imprecise concepts	Fuzzy controller for a plant irrigation system [37]
Learning Systems	Medium	Learning relationships among objects	Learning foraging behaviors in a robotic swarm [38]
Evolutionary Algorithms	Medium	Finding approximate solutions to challenging optimization problems	Calculating near-optimal path for data collection by a mobile sink [39]
Swarm Intelligence Algorithms	Medium	Finding approximate solutions to challenging optimization problems	Minimizing localization error of the sensors via a mobile anchor node [40]
Hybrid Systems	High	Combining the strengths of complementary techniques	Improve QoS metrics in a WSAN [41]

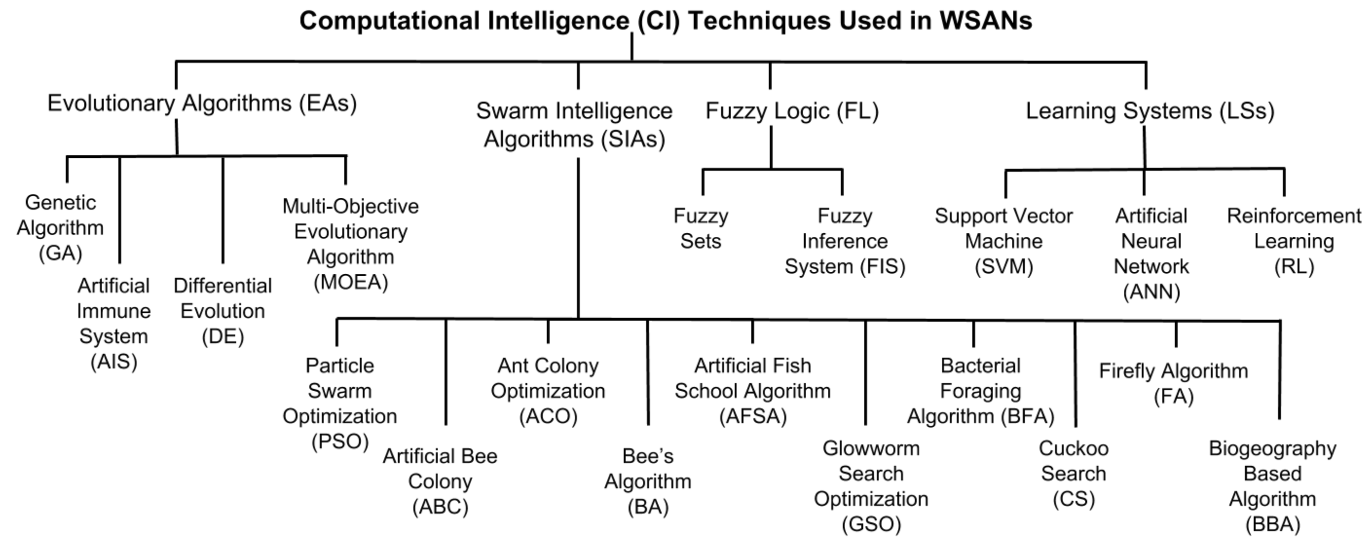


Fig. 1. CI techniques used in the surveyed WSAN papers

better individuals for particular situations. Finally, a single fitness function that must be optimized is used to evaluate the individuals. The best individual is returned as the best solution found.

A GA's performance is largely influenced by several factors, such as its parametric configuration. Additionally, there may exist hard or soft constraints that must be taken into consideration during the evolution. For example, a chromosome's genes ought not exceed certain bounds. Such individuals are deemed infeasible and must be either repaired or discarded. GAs are the most commonly found and popular type of EAs.

2) Multi-Objective Evolutionary Algorithms: (MOEAs) are an extension of EAs (particularly GAs) aimed at finding approximate solutions that are evaluated according to multiple, often conflicting objectives. The immediate difference between these techniques and classical, single-objective GAs is the fact that a single solution can rarely be given since an individual that optimizes a certain decision objective could worsen the others. Consequently, a tradeoff must be considered when optimizing one objective versus another. The solutions that do not strictly dominate each other form the *Pareto-optimal front* [44]. As such, MOEAs return a set of solutions that are part of this front, with another selection mechanism in place afterwards to make the final choice. One very popular MOEA is the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [45].

B. Swarm Intelligence Algorithms

Similar to EAs, Swarm Intelligence Algorithms (SIAs) are a branch of nature-inspired algorithms based on the interaction between living organisms [10]. If EAs are inspired by the biological phenomenon of evolution, SIAs's chief motivation is the collective behaviour of large animal groups, often referred as swarms. There are many commonalities between SIAs and EAs, for example they both maintain one or more populations of individuals and carry out an iterative optimization process guided by one or more fitness functions. However, SIAs work under the assumption that many individuals acting separately, but cooperatively, may be able to achieve a higher goal together (i.e., finding a good solution faster than on their own).

Like EAs, SIAs are not driven by gradient-based optimization principles. Instead they use other methods to explore the solution space. These algorithms are mainly exploited to discover combinations of values that optimize a certain fitness function in both combinatorial and numerical optimization problems.

Many algorithms that embody the SI principles have been put forth. Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) are two of the landmark SIA techniques

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

that will be presented in the sequel, along with Artificial Bee Colony (ABC).

1) Particle Swarm Optimization: PSO is based on the collective behavior of bird flocks and fish schools. First, a population of particles is created. Then, for each iteration, a particle's position and speed are updated relative to the position and speed of other elite solutions, namely the particle's own best position and that reported in its neighborhood. This algorithm attempts to explore the solution space by having groups of particles "swarm" towards high-value positions, hence discovering better solutions in terms of their fitness function. This process is repeated until some stopping criterion holds true. Similar to GA, the PSO algorithm has been adapted to cope with multi-objective optimization (MOO) problems.

2) Ant Colony Optimization: (ACO) is another well-known and representative SIA. This method emerged after the foraging behaviour of ant colonies. As the ants explore their environment, they deposit a chemical substance named pheronome on the ground. Upon encountering a food source, they head back to the nest while depositing pheromones in proportion to the quality of the food source. ACO was originally proposed for combinatorial (discrete) optimization problems. It first defines a set of solution components and pheromone values (the pheromone model). ACO works by probabilistically and iteratively assigning higher pheromone values to good solution components. These higher pheromone concentrations help the algorithm converge to promising regions of the underlying optimization graph. By varying the pheromone model, numerous ACO versions have been developed.

3) Artificial Bee Colony: (ABC) is another SIA technique that came about after observing the foraging behaviour of bee colonies [10], [46]. Individuals are categorized as employee bees, onlooker bees, and scout bees. When good food sources are discovered, employee bees stay at the food source while onlooker bees search near the source. Finally, scout bees explore the entire space. If no other food source is found near it, the employee bee switches to scouting mode. The technical details behind ABC can be found in [46]. Another algorithm based on the social behaviour of bees, the Bees Algorithm (BA), is discussed in [47].

4) Bacterial Foraging Algorithm: (BFA) is an optimization algorithm based on colonies of escherichia coli. The BFA algorithm is rather complex, and the details are available in [48]. It can be summed up as follows: a set of random individuals are generated. For each individual, chemotaxis as defined in [48] is carried out. Chemotaxis is the process by which an organism moves along the gradient of a substance concentration. Afterwards, the individuals will reproduce and split. A final step named elimination-dispersion is performed. A number of random individuals are eliminated, while a number of new individuals are randomly generated.

5) Artificial Fish Swarm Algorithm: (AFSA) is based on the social interaction of fish. The technical details are available in [49]. The algorithm follows five states with corresponding operations: preying, moving, swarming, leaping and following. Each of these steps attempt to either guide solutions towards high-value spaces, disperse to explore near local optima, or randomly explore the rest of the solution space.

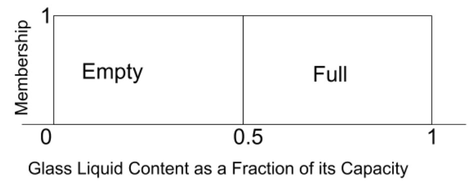


Fig. 2. Glass example - crisp logic

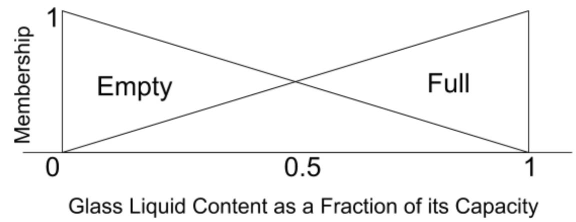


Fig. 3. Glass example - fuzzy logic

6) Cuckoo Search: (CS) [50] attempts to simulate the egg laying behaviour of cuckoo birds. Three main steps are performed in one CS iteration. First, an individual is created by modifying an existing individual, for instance via a Lévy flight [50]. The fitness of the newly generated individual is evaluated. This individual then replaces one of the individuals in the population. Next, some of the worst individuals are removed from the population. Finally, the best solutions are kept and ranked by their fitness function values. When a stopping criterion is reached, the best solution is kept and returned.

C. Fuzzy Logic

Fuzzy logic (FL) is a mechanism to reason in presence of vague and imprecise concepts [51]. Fuzzy logic imposes a membership degree of an object to a concept instead of simply concluding that an element either belongs to a concept or not, as done in classical set theory. A membership function maps elements in the universe of discourse to membership degrees between 0 and 1. FL is useful to quantify the vagueness permeating many real-world environments, which is a common trait of human reasoning. A half-empty, half-full glass will neither belong to the "full" or "empty" concepts in the crisp sense as shown in Figure 2 but will belong to both concepts with 50% membership degree under the FL interpretation.

In FL, we would have to first define the membership function for the "empty" and "full" linguistic terms of the "glass contents" fuzzy variable. A membership function is any function that assigns a membership value between 0 and 1 to an arbitrary input value in the domain of discourse. Popular membership function types are the Gaussian, triangular and trapezoidal functions. For the glass example, two triangular membership functions could be defined as shown in Figure 3.

It is now easy to see that both assertions are true. FL plays a pivotal role in many control systems and has been widely

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

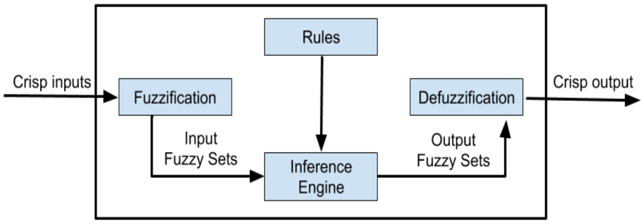


Fig. 4. Mamdani-type Fuzzy Inference System

hybridized with other Soft Computing techniques. Below we explain how a crisp value can be inferred from another set of crisp values by using FL.

D. Fuzzy Inference Systems

(FIS) are reasoning models that are able to infer a crisp value from a set of inputs, their fuzzy sets and membership functions, and a set of inference rules [52]. The two most commonly found FIS models are the Mamdani [53] and the Takagi-Sugeno FIS model [52]. The rules of an FIS are usually set via expert knowledge or automatically learned from available data. Such a system is generally built as follows: first, the system inputs (domain variables such as temperature or pressure) are fuzzified via the membership functions associated with the predefined fuzzy sets. The fuzzy sets are labelled by linguistics terms. The inference rules then use these linguistic terms as antecedents and/or consequents. The firing strength of each rule is then calculated, meaning that some rules could be more important than others in arriving at the conclusion. The firing strengths of all rules are aggregated and weighted, thus producing a fuzzy value. Finally, this value is defuzzified through appropriate methods like the centroid method for the Mamdani FIS, or the weighted average of the output values from the rules for the Sugeno model. A Mamdani FIS is portrayed in Figure 4.

E. Learning Systems

Learning Systems (LS) or *Machine Learning* is the process of uncovering relationships between a set of nominal/numerical features and states or objects [54], as living organisms do. This relationship is not known a priori by the system and must be "learned" from the available data. Three major LS avenues are generally recognized: unsupervised, supervised, and reinforcement learning. Each of these methods attempts to learn a model that infers the correct state of an entity as described by its set of features/actions. In *supervised learning*, a set of input features is mapped to one or more discrete object states (class labels). A supervised LS' task is then to discover this mapping between features and states from data. During the learning phase, the system is fed with a training set consisting of examples for which this feature-to-state mapping is known. The system attempts to optimize its internal model to have the highest number of correct inferences. When a satisfactory performance is attained, the system is used to infer the state of yet unseen objects. These systems are widely used for classification purposes. In

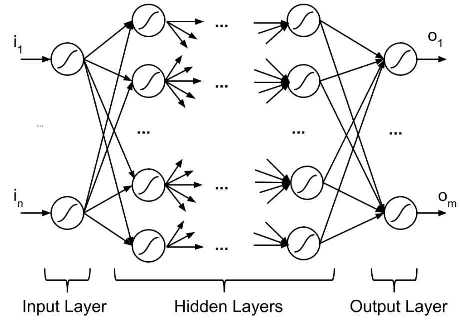


Fig. 5. Simple Artificial Neural Network

unsupervised learning, the system does not know the class labels (system state), but attempts to determine the relationship among certain features (association rule mining), and create states itself (clustering). Finally, in *reinforcement learning*, the system is fed with features, an inference is made, then the system is given a value proportional to the error in the inference. The system then attempts to minimize this error over time.

In the following we describe the two most prominent types of LSs that have been applied to WSANs.

1) Artificial Neural Networks: (ANNs) are loosely inspired by the brain, in which a set of neurons is able to process signals into the various outputs that regulate the body. A simple neural network is typically composed of interconnected neurons grouped into layers, e.g., the input layer, the output layer, and one or more hidden layers. Each neuron processes the incoming information from other neurons and applies an activation function, with the Sigmoid function, the Tanh function, and the Rectified Linear Unit (ReLU) function being some of the most popular choices. All edges between layers are weighted. A typical example of neural network is given in Figure 5.

These systems can be used in classification problems with the output being the set of class labels, in reinforcement learning as the policy function, in adaptive controllers where the outputs correspond to the control signals, and many more applications. The ANN research field quite vibrant and many breakthroughs, particularly in the Deep Learning arena, are being reported.

2) Reinforcement Learning: (RL) [55] [56] is a type of LS that determines an optimal policy dictating which actions to take at certain states in order to achieve the highest possible reward. The problem is often formulated as a Markov Decision Process (MDP), where there is a set of states and actions. The problem lies in discovering the optimal action-state associations, referred to as the policy function. A typical RL environment is given in Figure 6.

The interactions are as follows: the agent is in a given state and chooses an action from the set of actions, then receives a reward and transitions to a new state, where it chooses another action, and so on. Upon receiving a reward, the agent updates its action-choosing policy as per some learning function. In order to avoid continuously greedy yet suboptimal actions, the agent's policy usually integrates possible future

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

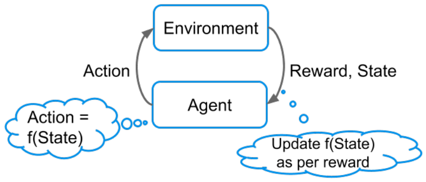


Fig. 6. Typical Reinforcement Learning Model

rewards for choosing an action. Finally, in order to avoid falling immediately into local optima, an exploration function must be defined, thus allowing the agent to efficiently explore the state/action joint space. Common RL algorithms are Q-learning and SARSA, with ongoing intensive research on new methods based on ANNs and Deep Learning [57].

Reinforcement learning algorithms are suitable in multi-agent systems where individual nodes do not communicate at all, or the social learning of all the individuals can be integrated to help these agents perform better.

F. Hybrid Systems

Hybrid Systems (HS) combine two or more types of elementary CI techniques in order to make up for the shortcomings of any one of the methods. For example, FIS rules require expert knowledge, but the task of finding rules that will yield the appropriate outputs is essentially an optimization problem. EAs and SIAs are suitable to solve such optimization problems. Consequently, many researchers employ EAs or SIAs to automatically learn the FIS structure (e.g, fuzzy rule base) [58].

Due to the similarity between EAs and SIAs, these two are often combined for greater synergy. Similarly, Fuzzy Logic and Neural Networks are sometimes combined, with fuzzy Adaptive Resonance Theory (ART) [59] being one example. We noticed a surge in the number of recent hybrid algorithms applied to WSAN scenarios.

IV. Wireless Sensor and Actuator Networks

Wireless Sensor Networks (WSNs) [60] are collections of static sensor nodes and one or more sink nodes. The often numerous sensor nodes are composed of one or more sensing modules, an energy source and a wireless communication device. The sensor nodes in these networks usually have limited computational power, thus requiring multihop chains to transmit their messages across the network to the sink nodes. WSNs are prone to node failure due to malfunction, energy depletion, malicious attacks or harsh environmental conditions. An extension of such networks is called *Wireless Sensor and Actuator Networks* (WSANs), which are made up of heterogeneous nodes capable of performing distributed computations and actuation tasks [11].

There are four key components to these networks: (1) the physical environment, (2) the sensor nodes, (3) the actuator nodes and (4) the sink node(s). Figure 7 gives a conceptual representation of a WSAN with these four components. Every WSAN will be acting within a bounded *physical environment*

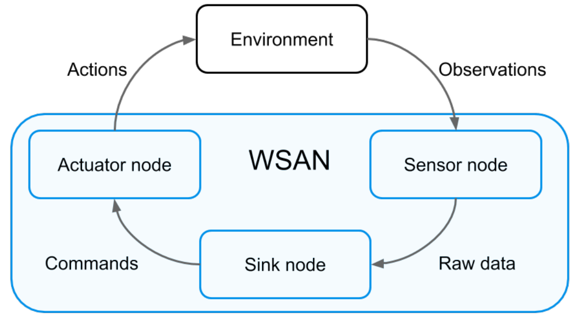


Fig. 7. Wireless Sensor and Actuator Network

of some sort, often called the Area of Interest (AOI). Then we have the *sensor nodes*, which are able to monitor the environment but not change its state. The nodes assuming this role could range from resource-rich mobile robots to cheap, static, resource-constrained sensors that are commonly found in typical WSN implementations. The *actuator nodes* represent the actionable network elements. Nodes assuming this role have actuation capabilities such as the ability to move, to pick up other nodes, or to act in a way that can change the state of the physical environment (e.g., a mobile robot, a light switch or a pressure valve). These nodes are typically resource-rich, computationally endowed, and more autonomous. They make either individually take or jointly coordinate their actions from sensor data (i.e., without any external input). Finally, the role of the *sink nodes* is to collect data from sensor nodes and to act as an interface to external agents that need to access/control the network. Similar to the actuator nodes, sink nodes usually have more computational power and energy, but may not act directly upon the environment. Nodes will be referred to as actuators, sensors, or sinks henceforth with the actual term depending on the context.

An important aspect of a WSAN is that a node may take on multiple roles at any time, if it is able to do so. For example, a node may perform both as an actuator and a sink, meaning that sensor nodes would forward data to it and some actions on the environment will be taken by this node. In other cases, a node might act both as a sensor and a sink, where a sensor is able to generate and preprocess data and send commands directly to the actuator nodes. Other joint roles are definitely possible depending on the application domain. A node may have all three roles, and be able to sense and act upon the environment without any external incentive. However, to be considered a WSAN, there must be many of these nodes that coordinate with each other. These types of networks are sometimes referred to as *Robotic Sensor Networks* (RSNs) [61] [62] or *Wireless Sensor and Robot Networks* (WSRNs) [5] [13] [14] [63], the latter implying that all the actuator nodes are exclusively robotic agents. A recently coined term is *Wireless Sensor, Actuator and Robot Networks* (WSARNs) [15], where other types of actuators co-exist with mobile robots.

There are many problems to be addressed in WSANs. For example, the communication task is harder than that in WSNs since the environment, and consequently the network topology, can not be deemed static. The task of controlling the topology to achieve a certain task (e..g, achieving a certain degree of sensor coverage) is in itself a difficult problem not found in WSNs since their topology is mostly considered static.

Five major types of problems have been identified as pertaining to WSANs, namely *actuation*, *communication*, *sink mobility*, *topology control* and *localization*. Each of these problems could be formulated in a variety of ways or may be further decomposed into several subproblems. We will study them in more detail in the next section.

V. WSAN Research Problems

This section discusses the main research problems formulated around WSANs that have been tackled through CI techniques. Figure 8 provides an overview of these challenges and their sub-problems. Some of these sub-problems have been already discussed in [11] [12] [13] [14] [15] but not from a CI perspective.

A. Actuation

The *actuation* problem is what separates WSANs from WSNs [64]. The actuation control loop must be closed, with data from sensors converted into actions upon the environment via the sink node(s). This conversion process itself can be relatively simple and rooted on traditional concepts from control systems theory or can also be complex and governed by advanced predictive abilities. The series of steps below will attempt to define the actuation control loop.

1) Task Creation: The first step in this process is *task creation* (TC). A task is a given action that must be taken by one or more actuators upon the environment, sometimes over an extended period of time. Task creation is concerned with the generation of proper tasks in a given scenario that allows for efficient actuation upon the environment. A task can be created either after an event has happened and been sensed by the sensors (reactive task creation) or an event could be predicted before it happens and tasks created accordingly (proactive task creation). This step often does involve data fusion. A task could be modified at a later time, given new information, though this substantially complicates the rest of the process. Task allocation is generally a context-dependent problem and hence cannot be placed neatly within a class of problems, though it must usually consider the unreliability of sensor data.

2) Actuator Selection: Following TC, actuators must be selected to enact them. *Actuator selection* (AS) is the problem of choosing the best subset of actuators that can solve a set of tasks. This step is often coupled with *task allocation* since it is generally more efficient to consider combinations of tasks and actuators together. However, it is sometimes better to perform a priori actuator selection to help reduce the number of possible actuator-task pairs, or to solve some of the constraints or objectives before trying to allocate the tasks. The interaction between actuators in the chosen subset and with the rest of the network should be considered; for that reason, this process is usually modeled as a combinatorial optimization problem. Information from sensors about the environment and related to actuators is not always reliable or even available, so these aspects should also be taken into account.

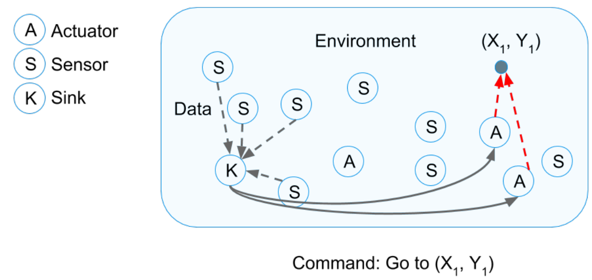


Fig. 9. Actuation control loop example

3) Task Allocation: Task allocation (TA) is the process of allocating one or more tasks to a subset of all actuators. A task may be allocated to one or multiple actuators, and any given actuator may by assigned zero or more tasks. The allocation should execute tasks with the goal of optimizing a certain objective and in a timely fashion. The problem is often called *Multi-Robot Task Allocation* (MRTA) [65]. Intuitively, this problem resembles that of finding an efficient assignment of tasks to actuators, and consequently is formulated as a combinatorial optimization problem. It is easy to see how the AS step can be integrated within the TA step.

4) Actuator Coordination: Following TA, agents work to resolve their tasks. Actuators may need the help of the network to accomplish their tasks, or they may need to coordinate with other actuators that have been assigned the same task. This is referred to as the *actuator coordination* (AC). The coordination problem is required throughout task resolution in order to avoid conflicts, and to have the most efficient process. This problem highly depends on the given task, therefore a domain-agnostic generalization cannot be guaranteed.

5) Event Prediction: A final step that is not always addressed is *event prediction* (EP). This is often conducted in parallel with the other steps. This problem requires the prediction of events that will influence the rest of the WSAN. For example, from sensor data, sink nodes may have systems that can predict when an event will happen. A task can then be created beforehand to mitigate this event. Prediction is a central concept of proactive systems. Predictions can be solved by LS and to a lesser degree via FL.

Any actuator or sink node may be carrying out one or more of these steps at any given time. For example, a sink node may be creating tasks, allocating previously created tasks, and predicting new ones, all in parallel. Similarly, an actuator may be executing one of its tasks while being allocated new ones simultaneously. An actuator may even create new tasks from one of its allocated tasks in order to accomplish it, hence spawning another nested process. Figure 9 shows an example of a sink node collecting data from multiple sensors, then creating a task to explore a certain region of the AOI. It selects two of the three actuators, and allocates each the task of exploring the monitoring region.

B. Communication

Communication issues are one of the major challenges in WSNs, with many attempts to mitigate them being the focus

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

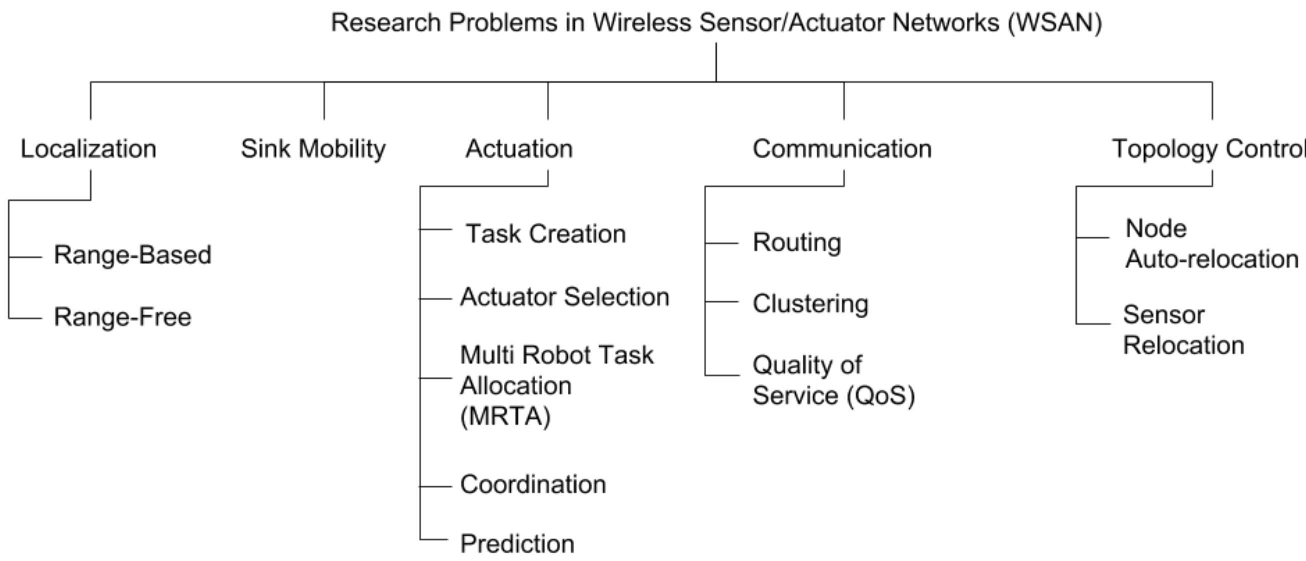


Fig. 8. A taxonomy of research problems in WSANs

of active research [66]. WSNs are usually self-configuring systems, meaning that the network must configure itself efficiently to be able to route messages to the sink node. This is a hard problem to solve for a WSN since such a network is generally composed of lightweight, energy-constrained nodes. Consequently, any communication protocol must take this into consideration. For example, there is a cost associated with the repeated use of a communication route since it drains the energy of those nodes on that route.

As in many other cases, this problem is exacerbated in WSANs due to the mobility of the actuator nodes. As the network often exhibits a dynamic behaviour, pre-existing communication protocols are not always applicable, especially in the case of a mobile sink. The addition of actuators is not bad at all, since it integrates nodes with higher additional computation and actuation resources. Moreover, the actors' capabilities may be leveraged directly, such as relocation of sensor nodes to restore connectivity, or an actor moving itself to act as a bridge between two otherwise disconnected subnetworks.

The WSAN communication problem is quite broad and many sub-problems have been identified over the years. The following subsections will elaborate on the three main sub-problems, namely routing, clustering and Quality of Service management in WSANs.

1) Routing: The *routing* problem in WSANs shares many commonalities with the traditional routing problem in WSNs. However, it is more constrained due to some of the nodes' limited abilities and resources. An important problem in WSNs, named the *sinkhole problem*, happens when many messages are routed to the sink node. Since these messages end up overusing some of the multi-hop communication routes, a few nodes closer to the sink get rapidly depleted of their energy, thus leading to the sink node being unable to communicate with the rest of the network. This type of situation may happen anywhere in the network, so a good routing protocol must take the residual energy into consideration [67] [68].

Additionally, sink nodes may move through the network, with previously used network routes not being relevant anymore. In fact, one of the solutions to the *sinkhole problem* is the use of a mobile sink. However, as this sink makes its way through the network, existing communication routes must be re-established without incurring in significant latency, energy expenditures or throughput degradation. Finally, as actors may behave as sink nodes themselves in some WSAN implementations, and given that there might be multiple actors, the routing protocols are not limited to routing to one sink node, as they may choose any actor to route the information to.

2) Clustering: Another approach is to create structures in the network for more efficient communication [69]. The most common structure is the cluster, i.e., a tree with the root node being called the cluster head. These clusters create a hierarchy in the network that simplifies other tasks such as routing, since only cluster heads must handle routing protocols to discover efficient routes to the sink node.

The choice of cluster heads must be carefully made. Since the cluster heads will be frequently engaged in communication tasks, their energy levels will deplete faster than the cluster members. Consequently, clusters are ephemeral since once a cluster head is low on energy, the clustering algorithm must be rerun in order to elect another node as a cluster head, most likely one that has enjoyed lower energy expenses due to simply being a cluster member. When applicable, actors make good cluster heads due to their higher energy availability, computational power, and often longer communication ranges.

3) Quality of Service: Quality of Service (QoS) management is the task of controlling the network resources so as to ensure high throughput, low error rates, minimal number of dropped packets, and low communication latency [41]. Such network indicators are relevant to WSANs, where data gets aggregated to actors, and actors must often coordinate among

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

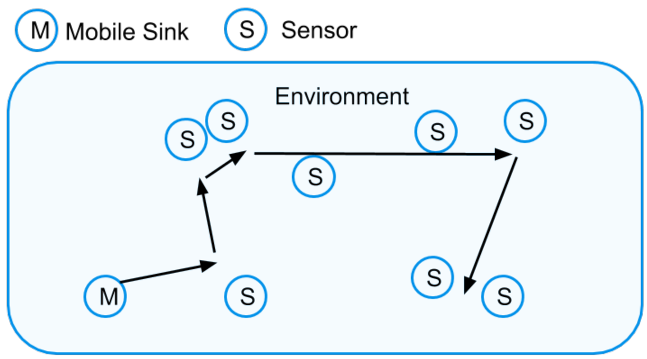


Fig. 10. Sink mobility example

themselves to solve complex tasks where a low latency is required.

C. Sink Mobility

One of the earliest WSAN motivations was that of efficient routing to the sink node for data collection. Since all data is routed back to the sink node in WSNs, the nodes closer to the sink would continuously be forwarding messages, thus quickly depleting them of energy [70], the so-called *sink hole problem*. Efficient routing algorithms that would generate alternate routes were devised, while others developed WSNs with mobile sink nodes, thus transitioning the network from a static WSN to a WSAN with mobile nodes.

The traditional sink mobility problem in WSANs can be formulated as follows: given a network of static sensors, find the optimal data collection trajectory for a mobile sink that is efficient in terms of communication, distance travelled, and energy utilization. This means the sink node must travel the shortest distance while visiting enough static nodes to gather all information to be transmitted, and use a path that minimizes energy usage. Most algorithms assume the sink node to be powerful, therefore they concentrate on preserving the sensing nodes' energy efficiency. Figure 10 illustrates this traditional problem.

Other variations of this problem include the presence of multiple mobile sinks and the addition of locomotion to the sensor nodes themselves, as will be discussed in Sections VII-C and VIII-D.

D. Topology Control

Topology control is a fundamental WSAN problem [11]. Whereas WSNs exhibit static network topologies, actuators in WSANs have the ability to improve sensor coverage by relocating sensor nodes, deploying new nodes, or moving themselves in case extra sensing capabilities are required. The coverage-based topology control problem can be formulated as follows: given a bounded n-dimensional space, and p n-dimensional spaces already located within the bounded space, relocate the p spaces or add additional n-dimensional spaces in such a way to increase the proportion of the sum of all p spaces over the bounded space. Note that n must be greater than zero and p greater than or equal to zero. This is a rough definition of coverage that attempts to optimize 1-redundant coverage over the network. However, it could be extended to k-redundant coverage as well.

Communication in a WSAN is known to be unreliable. Modifying the underlying network topology is a direct factor contributing to this unreliability. Consequently, the presented methods must not only consider sensor coverage but also the communication impact brought about by the topological change. Similarly, mobile actuator nodes might drain static nodes of their energy if frequent communication is established, thus reducing the network lifespan and hence the amount of time the network will be able to monitor the AOI. For these reasons, topology control solutions often include some aspect of communication or MRTA, hence bringing their own challenges.

WSANs are made up of heterogeneous nodes that must seamlessly interact with each other when reshaping the network topology. Actuators with sensing capabilities (or the sensor nodes themselves if they are mobile) may position themselves at optimal locations to repair or augment coverage, or they can either relocate or deploy static sensor nodes. Both of these perspectives require careful planning from the sensor nodes, the actuator nodes or both and each case must be approached differently. We elaborate on them in the following subsections.

1) Sensor Relocation: This refers to the task of moving one or more sensors from one location to another in the WSAN in order to serve a particular goal, e.g., restore/augment the network coverage or better support data collection efforts. Two avenues have been pursued within sensor relocation: (a) **self-relocation** [11] and (b) **actuator-assisted relocation** [13]. The former assumes that the sensor nodes are mobile and hence may relocate themselves. This might result in higher communication disruptions due to the topology of large areas of the network changing simultaneously. Additionally, relocation of mobile nodes is a more expensive solution since many nodes, if not all, must be equipped with locomotion capabilities, which also rapidly drains the nodes' energy. This brings about a trade-off among sensor coverage, network cost and network lifespan. However, node self-relocation is one of the quickest and easiest solutions developed to restore sensor coverage, since the WSAN can effectively reorganize itself. Figure 11 showcases this problem.

The alternative to sensor self-relocation is actuator-assisted relocation. In this solution, the sensor nodes are generally static and their relocation is entrusted to a few actuators that may pick them up, carry them along the network and drop them off at the intended positions. This solution is cheaper and more realistic than mobile sensor relocation. Figure 12 illustrates the scenario where an actuator relocates a sensor by picking it up and dropping it off at a new position.

2) Sensor Deployment: A second perspective is the deployment of sensors by actuators [11]. The goal here is to repair or augment sensor coverage by deploying new sensors in the monitoring region. Contrary to sensor self-relocation, the deployment strategy is entirely governed by the actuators, which decide (either individually or corporately) if additional

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

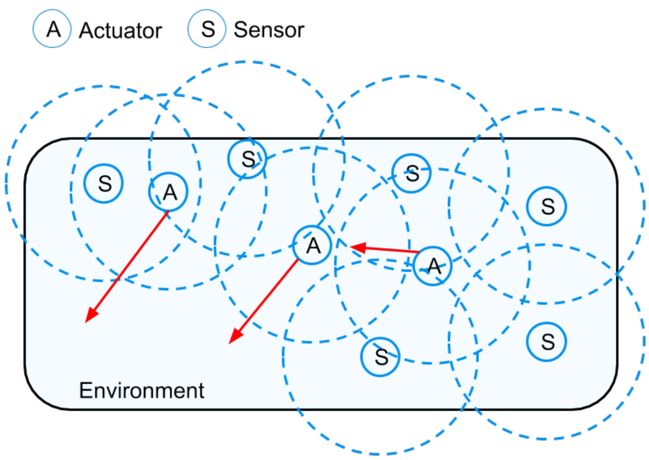


Fig. 11. Node self-relocation example

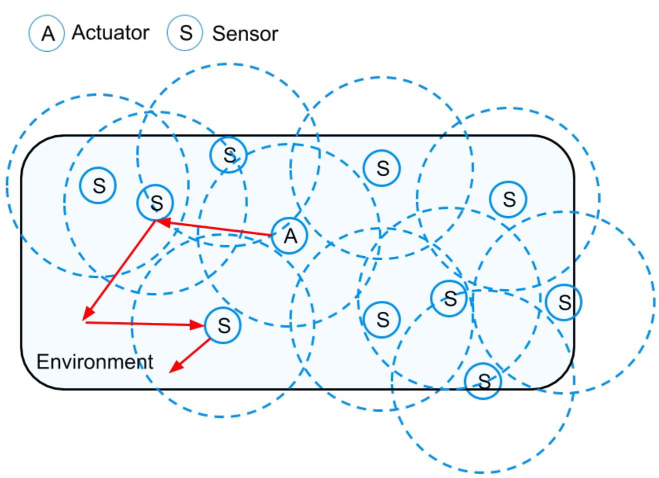


Fig. 12. Actuator-assisted sensor relocation example

sensors are needed, where to deploy them and which route to pursue so as to achieve all network coverage goals. Most of the studies assume that the actuators are able to carry an unlimited number of sensors, which is clearly unrealistic.

Deploying sensors by actuators has the benefit of being a less costly strategy due to the network topology remaining fairly static and closer to that of a WSN. This makes communication and energy usage less of a concern since the actuators are deemed to be resource-rich. However, given that the actuators must plan the best deployment strategy considering the state of the rest of the sensors in a region, the deployment of the new nodes might take some time. A similar remark could be made about actuator-assisted sensor relocation if the planning takes place in a centralized fashion.

3) Sensor Replenishment: This is a recently promising topic in WSAN research that has seldom been explored from a CI perspective. The idea is that one or more actuators are able to replenish the energy source (oftentimes a battery) of the static sensors they come in contact with [71] thanks to the recent breakthroughs on wireless power transfer between devices [72] [73]. This approach carries the potential to make the WSNs immortal in a sense, for no sensor node will

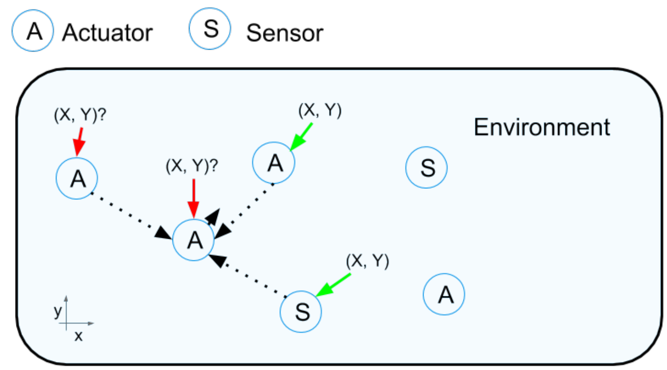


Fig. 13. Localization example

ever run out of battery if the replenishment cycles are duly planned [74]. The topological arrangement of the WSN in this case is not about changing the location of the nodes but in boosting their energy availability. The literature often refers to this kind of WSAN as a *Wireless Rechargeable Sensor Network* (WRSN) [75]. Militano et. al. [76] recently published an interesting discussion on the advantages and limitations of recharging versus replacing sensor nodes by mobile robots.

E. Localization

Knowing a node's location in a WSAN is not always easy. As with many problems in WSANs, the localization technique must make a compromise between energy efficiency, cost and accuracy [77] [78] [79]. The easiest solution is equipping every node with a Global Positioning System (GPS) device, though this is often too costly to be practical and would quickly drain nodes of their limited energy, thus severely limiting the lifespan of the network. Furthermore, GPS devices are not always reliable if available at all, such as in indoor environments or urban areas. Consequently, a better solution must be devised. All solutions require some nodes to be aware of their precise position in a given global reference frame, or else it would be impossible to localize a WSAN. Such nodes are often referred to as *anchor nodes*, and may be special sensor nodes, actuator nodes, or the sink node. This problem can then be formulated as such: provide an accurate and efficient localization service to nodes, given that one or more nodes know their precise location [80]. Figure 13 exemplifies this problem, where a mobile actuator attempts to determine its own location.

Anchor nodes know their position either via direct initialization or through GPS. The localization problem can be approached from multiple angles. Some methods attempt to gain meta-information from communication with other nodes, such as time-of-arrival (TOA), angle-of-arrival (AOA), or received-signal-strength (RSS/I); methods that use such features are classified as *range-based* localization techniques. The alternative, called *range-free* localization, localizes nodes using the broadcast positions of the anchor nodes. This kind of localization algorithms do not attempt in any way to determine the range of the message.

While some localization methods developed for WSNs may be appropriate for WSANs, more refined schemes can be devised for networks with actors, especially when the nodes are mobile.

VI. CI Techniques Applied to WSAN Problems

This Section break down the application of CI techniques into the five major WSAN problem categories outlined in Section IV and their sub-problems. It concludes by summarizing the main research findings along each problem category.

A. Actuation

As depicted in Table II, most of the surveyed papers do not cover the five actuation subproblems outlined in Section V-A but focus on some and omit others depending on the scope and characteristics of the problem they are trying to solve. However, we do see two clearly differentiating groups of CI-based actuation approaches in WSANs: those addressing *early actuation* efforts such as task creation, actuator selection and task allocation, and those concerned with *late actuation* steps, which are mainly related to task execution (via actuator coordination) and event prediction. It is worth mentioning that we only highlighted those actuation subproblems that were the main subject of attention in every reviewed paper.

1) Early Actuation - Focus on Task Allocation: The authors of [81] first proposed a GA-based cluster creation algorithm, then presented a PSO-based method to relocate this cluster as an entity. The work focuses on task allocation, where a given task would be to move the network to a certain area, the allocation would assign tasks to nodes in the network in such a way as to not lose cluster coherency. The method is demonstrated by simulations but not compared to other techniques.

An interesting WSAN application is unveiled in [114], where the WSAN serves as a Decision Support System (DSS) to guide occupants of a building towards the exits. In this network, motion sensors are used as sensors and light switches as actuators. The network leans on an ANN to allocate light on or off tasks to the actuators, and learns to predict these tasks during the training phase. The network also learns which paths are optimal to the exits. Then, in emergency scenarios, the light switches activate exit signs, hence guiding building occupants towards the exits. The method is validated by a simple simulation.

In [37], [100], [101], a WSAN for controlling an agricultural environment such as a greenhouse is put forth. In these works, sensors detect humidity levels, temperature, or other indicators of the current environmental condition of the greenhouse. This data is then fed into an FIS that translates it into applicable tasks for the actuators that are made up of components that can influence this environment, like ventilators or irrigation control systems. The authors in [37] proposed an architecture for automatic irrigation control. A simulation is presented in [100], and a later work by the same authors [101] demonstrated a physical implementation and then conducted tests that showed the capabilities and feasibility of the system.

An auction-based protocol to select the best actuators is presented in [61]. Following a task creation, an auction is triggered, where actuators compute their bid from their energy levels, distance to the event and current redundant coverage via an FIS. Then, the top bidding actuators get selected and allocated tasks to recover coverage. The task allocation process is formulated in a multiobjective optimization (MOO) fashion and driven by NSGA-II. The method is then validated via simulations. This framework is used for Critical Infrastructure Protection in [120].

The work of [62] builds upon [61]. It concentrates on the actuator selection and task allocation stages. The authors propose to first create a task that is related to coverage restoration. Then, this step is optimized by the use of a MOEA to determine sub-tasks through the use of an actuator selection step. In the former, optimal positions for actuators are determined, while in the latter, agents utilize an FIS to determine their bid in an auction for task participation, a market-based technique. The top bidding actuators then become the subset of actuators that may get allocated a task. If the MOEA divides the task into sub-tasks, an allocation step is carried out, where robots bid on one of the sub-tasks by using an FIS that determines the agent's ability to accomplish any given sub-task. If the original task was not subdivided, an actuator subset is created. This subset, called the coalition, is fed into a MOEA that allocates sub-tasks on the basis of reducing energy usage and maximizing restored coverage. Both techniques are presented and then simulated in various network scenarios to demonstrate their effectiveness.

The authors in [102] described a WSAN application in lighting control. This work focuses more on task creation. Sensor data is sent to an FIS controller that creates tasks for specific actuators. No simulation of the work was given.

In [83], a WSAN is used to track a target. The sensors gather information about the target, then a prediction of its next position is estimated. A GA is then used to select which mobile sensors to relocate in order to cover the predicted path of the target. This GA considers both actuator selection and task allocation simultaneously, hence optimizing the time needed to catch up with the tracked object, and the distance to the tracked object. The work is validated by a simulation.

A MOEA-based task allocation protocol for sensor replacement by multiple robots is brought forth in [88]. In this work, sensor drop-off or pickup tasks get created and allocated by the MOEA algorithm. The optimization considers four conflicting objectives: (i) minimizing the total length of the sensor relocation trajectories traveled by all robots; (ii/iii) maximizing the robustness/lifetime of those trajectories and (iv) minimizing the load balancing factor among all robots. Multiple MOEA techniques are compared in the simulations. This work is an extension of [86] and [87], which addressed the single-robot case.

The authors of [84] elaborated on another interesting WSAN application to automate lighting control in media production. They developed a GA-based method where actuators appropriately increased the luminosity for media production based on sensor readings. They also identify positions where additional sensors could be deployed to increase accuracy. The method

TABLE II
APPLICATION OF CI TECHNIQUES TO THE ACTUATION PROBLEM IN WSANS

CI Technique	Subproblems	Algorithm	Reference	Validation	Computation Distribution
Evolutionary Algorithms	TC,AS,TA	GA	[81], [82]	Simulation	Centralized
	TC,AS,TA	GA	[83]–[85]	Simulation	Centralized
	TC,TA	MOEA	[86]–[88]	Simulation	Centralized
	TC,AS,TA	MOEA	[89]	Simulation	Centralized
Swarm Intelligence Algorithms	EP	ACO	[90]	Simulation	Centralized
	TC,AS,TA	PSO	[91], [92]	Simulation	Centralized
	TA	PSO	[93], [94]	Simulation	Centralized
	TC,AS,TA	PSO	[95]	N/A	Centralized
	AS,TA	ACO	[96]	Implemented	Centralized
	TC,AS,TA	ABC	[97]	Simulation	Centralized
	TC,AS,TA	ACO	[98]	Simulation	Centralized
	TC,AS,TA	PSO	[99]	Implemented	Centralized
Fuzzy Logic	TC,AS,TA	FIS	[37], [100]	Simulation	Centralized
	TC,AS,TA	FIS	[101]	Implemented	Centralized
	TC,AS,TA	FIS	[102]	N/A	Centralized
	TC,AS,TA	FIS	[89]	Simulation	Centralized
	AS,TA,AC	FIS	[103]	Simulation	Distributed
	AS	FIS	[104]–[111]	Simulation	Distributed
	TC,AS,TA,AC	FIS	[112]	Simulation	Distributed
	AC	FIS	[77], [113]	Simulation	Centralized
Learning Systems	TC,AS,TA,EP	ANN	[114]	Simulation	Centralized
	EP	competitive Hopfield ANN	[115]	Simulation	Centralized
	TC,AS,TA	Artificial Endocrine System with ANN	[38]	Simulation	Distributed
	TC,AS,TA,AC	ANN	[116]	Simulation	Centralized
	TC,AS,TA,AC	ANN	[117]	Implemented	Centralized
	AC,EP	RL	[118], [119]	Simulated	Distributed
Hybrid System	TC,AS,TA	FIS/MOEA	[61], [62], [120]	Simulation	Centralized
	TC,AS,TA,EP	FL/ANN	[121]	Simulation	Centralized
	AS,TA	FIS/MOEA	[87]	Simulation	Centralized
	TC,AS,TA,AC,EP	EA/SIA	[122]	Implemented	Centralized/Distributed
	TC,AS,TA	FIS/PSO	[95]	N/A	Distributed
	TC,AS,TA	FIS/PSO	[112]	Simulation	Centralized
	TA,TC	FIS/ANN	[113]	Simulation	Centralized
	TC,AS,TA	ACO/ANN	[123]	Simulation	Centralized

is validated by simulation.

In [91], a TC-AS-TA method is described. When events are triggered by an adaptive distributed event, a task is first created, the number of actuators needed is then considered and chosen, and the tasks are allocated by a modified PSO scheme. The paper approaches these concepts from a mathematical point of view grounded in control theory. Their method, called ADET, significantly outperforms the non-adaptive version in the simulation analysis.

An MRTA solution is presented in [93]. A PSO-based method is used to allocate tasks to a heterogeneous set of actuators. The method is simulated and demonstrates that it can converge to an appropriate solution.

An approach to schedule time-delayed tasks is laid out in [82]. More specifically, the authors employed a GA to generate an actuator allocation schedule that reduces the peak electricity use. The method is then validated by experimentation.

In [104]–[111], actuator selection is carried out through an FIS. First, a state model for the actuator is defined by taking into consideration the method's ability to resolve the task. Then, the FIS maps an actuator's state onto a numerical value indicating the actuator's suitability to complete the task. The most capable actuators are then selected for task allocation purposes [105]. The study in [104] considers four input variables to the FIS, namely, the type of required action, the distance to event, the remaining power and the security of the task request. The work in [106] is a continuation of what was accomplished in [105], where the reliability of the information was considered. In [107], a different model that focuses on the actuator's ability to efficiently and quickly resolve a task is unveiled. Next, [108] actively considers the actor-sensor coordination quality as an additional input to the FIS for actor selection. The work in [109] replaces the latter input variable with the failure of assigned task. Finally, [110] and [111] investigate the effect of actor node density upon the selection of the actor nodes. All these methods are validated by a simple simulation analysis.

The authors of [38] disclosed a novel framework for WSAN control based on an Artificial Endocrine System (AES) augmented with ANNs. An AES is a control framework that borrows concepts from endocrine systems found in humans and animals. Goals are defined in the context of the AES, then the AES releases either positive or negative feedback that influences the actuators, thus guiding the WSAN to complete the defined goals.

A swarm navigation algorithm is presented in [95]. This algorithm efficiently creates and allocates tasks to search for a target in an AOI based on two CI techniques: a PSO algorithm and an FIS whose rules have been optimized by PSO. The paper does not include an end simulation but validates some of its concepts.

An MRTA framework is put forward in [96]. The authors define a base set of abstract tasks from which other tasks can be constructed. Then, an adaptive ACO method is utilized to discover an efficient TA method to a subset of actuators. The scheme is validated through real experimentation that showed its feasibility in real-world scenarios.

A target tracking method is presented in [97], [98]. This approach first uses sensor data to predict future locations of a target. Then, in [97], an ABC is used to allocate relocation tasks to mobile nodes in order to cover the projected area while [98] makes use of an ACO algorithm. The methods are validated by simulations.

In [99], an AOI exploration method is researched. This method is quite simple: it uses PSO literally, where each mobile node corresponds to a particle. A real-world scenario was devised to evaluate the method; it demonstrated its potential and feasibility.

The AOI search method put forward in [85] envisions a GA in charge of determining suitable search points and allocating relocation tasks to a set of mobile nodes. The method is validated by simulations.

Another search method was published in [92]. In this work, the AOI is first split into a grid. Then, mobile nodes randomly explore a cell. Once a cell's search percentage goes above a threshold, they move to another cell. When a mobile node uncovers an object of importance, a PSO algorithm is run to either instruct mobile nodes to stay in their current cell, move to another area, or help the mobile node that uncovered the object in searching its cell. The method is validated via a simulation study.

An MRTA process is described in [123], where the probability of an actuator completing the given task is considered. The MRTA process is embodied through a modified ACO algorithm while the probabilities are estimated by an ANN. The method is simulated with real robots and the empirical results are provided.

The study in [77] is concerned with the task creation step. In order to determine their next position while searching an AOI, mobile sensors use an FIS whose inputs are the mobile nodes' positions and outputs the next optimal positions by using Swarm Intelligence principles. The work is simulated and shown to be more efficient than a gradient-based approach.

The authors of [94] present an MRTA for WSANs based on a modification of binary PSO, a particular case of PSO in which each particle encodes a sequence of bits. They formulated task workload and connectivity as constraints while optimizing task execution time, energy use, and network lifetime. Their method was finally validated by simulations.

Primeau et. al. [89] envisioned a tight coordination between a UAV network and a ground network of potential responders in order to mitigate maritime smuggling operations. Upon detection of a potential smuggling, the UAV network self-organizes through a MOEA to corporately confirm the existence of such operation. The risk-driven analysis for smuggling detection is powered by an FIS using the Risk Management Framework in [124] [125]. Once the event has been confirmed, the UAV network alerts the ground network, which proceeds to decide how to best respond to this illicit activity, again by leaning on different FIS and MOEA implementations.

2) Late Actuation - Focus on Task Execution and Event Prediction: Many of the above control/actuation process steps are found in [121], where a WSAN is used to detect abnormalities. In this system, a Fuzzy Adaptive Resonance Theory (Fuzzy-ART) neural network aims to detect abnormal behaviour based on a Markov Chain model. A Fuzzy-ART system is a LS that combines FL and ANN in an attempt to mimic human reasoning. The system is first trained. Then, from sensor data, the network predicts abnormal behaviour, therefore creating the task of sampling a specific region for more information on the abnormality. An actuator is selected and assigned the task. This method is validated by experimentation.

First presented as a topology control problem, [90] actually proposes a method for optimal path planning in WSANs, an important part of task resolution and coordination. The method is a modified ACO algorithm, with mobile nodes acting as the ants. They present a simulation that shows nodes attempting to find the shortest route to a point.

A target tracking problem in networks of multiple mobile nodes is presented in [115]. This algorithm could be considered to be part of the prediction step, since it attempts to create tracks for the tracked targets, which can be later used to create efficient tasks. The proposed solution relies on a Competitive Hopfield Neural Network (CHNN) to identify multiple targets and build tracks from information sent by multiple mobile sensors. The work is simulated and the results are discussed.

In [103], the authors investigated a coordination strategy for mobile nodes. Positions are determined in order to create a certain formation, then an FIS is used to ensure that nodes do not collide with each other. This work falls within the task execution through multi-agent coordination category. The method is validated via simulations.

The study in [122] is a collection of ongoing efforts to create a WSAN swarm that closely resembles their natural inspiration at many levels. In the work, many concepts such as self-organization of many nodes to act as an entity, task allocation, and many more advanced ideas are solved by AIS, complex EAs and ANNs. All mentioned works are in progress, and [122] only gives a glimpse of what can be attained. There are no experimental validations given, though many figures demonstrate the intriguing yet promising concepts.

A control system for WSAN is defined in [112] for search and exploration in the AOI. This work pitches two methods: the first one based on a PSO engine augmented with FL, and the second method rooted on an FIS. The first searches the space guided by the fuzzy PSO, while in the second method, mobile nodes evaluate their fitness, and nodes are attracted to the node with the best fitness.

A coordination mechanism to resolve conflicting actuator actions is found in [113]. An FIS is implemented in actuators and is used to complete the task resolution process. The method was tested on a real setting where nodes needed to cooperate to push a box and the method enabled the completion of the task.

In [116] [117], an intriguing behavioural system is presented. A method that allows for behaviour to emerge instead of being rigorously defined is introduced. First, the objective functions are defined. Then, the robots' ANN-based controllers

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

learn from interactions with the environment. The authors proposed two methods of learning, namely individual and social, where social learning allows nodes to learn from other nodes' experience. The goal is to have online emerging cooperative behaviour from a few simple objectives. The method is demonstrated in a foraging or exploration/search scenario. The actuator controllers are first trained through simulation, then they are deployed on actual physical agents that show better performance than a random controller. The paper in [117] is an extension of [116], where the method is augmented and tested on four different scenarios in an aquatic environment.

The authors in [118], [119] tackled event prediction and agent behaviour in WSANs with some mobile nodes. Events are first predicted individually by actuator nodes through a maximum likelihood estimation (MLE) approach using spatial correlation in the sensor data. An MDP is then defined to control the behaviour of mobile actuator nodes, guiding it towards the event. In this process, the actuators nodes must decide which node within its communication range to visit. The MDP can be solved using RL, with the actuator being rewarded for moving to a node with a higher MLE estimate. In [119], this method is enhanced by adding an exploration behaviour inspired by desert ants. Both methods are validated through a simulation analysis.

B. Communication

This section presents relevant work for all three subcategories of the WSAN Communication problem. Table III outlines all surveyed works in this category. From a CI perspective, this area is still a fairly untrodden territory.

1) Routing: All five uncovered approaches consider routing to one or multiple mobile sinks. Most of them aim at optimizing one or more features of the routing paths. One approach puts forth an interesting idea where the optimization applies to the mobile sink trajectories and the ensuing routing paths are a by-product of this step.

Routing Path Optimization

The authors of [134] introduce a routing algorithm to discover communication paths to a mobile sink node. This algorithm maintains routes through a hybrid AIS-ABC, where ABC's role is to find route solutions, and AIS is responsible for generating new immunized solutions. This algorithm uses a fitness function that takes into account signal strength, latency, and energy use. The proposed approach is simulated and compared to two non-CI methods.

In [127], the authors created a routing protocol for mobile sinks termed SIMPLE by employing a modified PSO algorithm. SIMPLE discovers routes with higher residual energy. The authors then simulate their proposed method for a variety of sink speeds and message bandwidths. They conclude that sink speed does not have as big an effect on network lifetime as message bandwidth. Two versions of the algorithms are compared to two other routing protocols; the results corroborate that SIMPLE performs better.

A routing protocol to multiple mobile sinks that optimizes minimal delay, route distance and energy expenditures along the route is put forth in [128]. The method uses a cooperative PSO algorithm enhanced with endocrine principles to increase its convergence and exploration capabilities. The proposed work was simulated and compared against two other algorithms; the experimental evidence indicated that the new algorithm outperforms its peers in some aspects. In other indicators, like delay, the algorithm did not do as well in small networks, but scaled much better.

Another routing protocol to multiple mobile sinks is developed in [129]. This protocol leans upon a modified PSO algorithm that incorporates greedy techniques with a memory component. This memory component stores previously searched solutions to avoid duplicating them, thus speeding up convergence. The algorithm optimizes energy usage and communication delay. The proposed algorithm is simulated and compared against three other methods, one of them being a similar PSO scheme, with the proposed work outperforming the latter.

Mobile Sink Path Optimization

The study in [130] unveils a different routing algorithm for mobile WSANs by having PSO optimize the mobile nodes' trajectories in such a way to construct dynamic communication backbones. Instead of uncovering appropriate routes through a network, the network itself dynamically changes its topology to construct the backbone. Such a method is demonstrated in the work as a simulation of a UAV swarm shows how routing bridges are formed.

2) Clustering: The clustering methods reviewed in this section take into account the mobility of the nodes in the WSAN and aim at optimizing the final clustering of the network in terms of the number of cluster heads and other performance metrics.

The method in [81] generates clusters in mobile WSANs via an EA, then studies how to coordinate the movement of clusters using PSO. Only the clustering aspect of this work is considered here. The GA operates over the entire network and tries to optimize a fitness function consisting of the number of cluster heads and the distance to the cluster heads from other cluster members. These two criteria are weighed to account for their relative importance. The method is then simulated, showing its effectiveness.

Srivastava and Sudarhan [135] elaborated on a two-step process to determining clusters and cluster heads in mobile WSANs. The first step is for each node to determine its suitability to become a cluster head via an FIS whose inputs are the distance to other nodes, the remaining energy, the local node density, and the node mobility. Afterwards, a GA is charged with making the final selection of cluster heads. Each candidate solution is evaluated in terms of its number of cluster heads, the mean communication energy required and the speed of the cluster heads. The simulations demonstrate the viability of the proposed method across different FIS settings.

The authors of [126] put forward a centralized GA-based algorithm named GAROUTE that clusters WSANs while taking the mobility of the nodes into consideration. The GA determines the best cluster heads to form 1-deep clusters by optimizing the mean communication energy, the number of cluster heads, and the total cluster head speed. Whenever a node exits a cluster, it forms its own cluster. Simulation of the method in comparison to other clustering algorithms

TABLE III
APPLICATION OF CI TECHNIQUES TO THE COMMUNICATION PROBLEM IN WSANS

CI Technique	Subproblem	Algorithm	References	Validation	Computation Distribution
Evolutionary Algorithm	Clustering	GA	[81], [126]	Simulation	Centralized
Swarm Intelligence Algorithm	Routing	PSO	[127]–[130]	Simulation	Centralized
Fuzzy Logic	Clustering	FIS	[131]	N/A	Centralized
Learning Systems	QoS	FIS	[132]	Simulation	Distributed
		ANN/RL	[133]	Simulation	Distributed
Hybrid System	Routing	AIS/ABC	[134]	Simulation	Centralized
	Clustering	FIS/GA	[135]	Simulation	Centralized
	QoS	FIS/GA	[41]	Simulation	Distributed

demonstrates that GAROUTE does result in a lower energy consumption.

Fuzzy-CEACH (Configurable, Energy-Efficient, Application-Aware, Channel-Aware, Clustering based Service for WSAN) is introduced in [131]. CEACH is a communication middleware service that handles the task of providing reliable QoS communication in Cyber-Physical Systems, and WSANs in particular. One of the CEACH tasks is cluster creation. An extension of this middleware, Fuzzy-CEACH relies on an FIS to determine the appropriateness of a particular node to become a cluster head. This is a function of several indicators, including the node's residual energy, connectivity degree, in/out link reliability and hop-distance from the sink. The authors do not provide any Fuzzy-CEACH results though.

3) Quality of Service: In [132], Xia et. al. focus on providing a target QoS for sensor-to-actuator communication. This scheme deals with the impact of unpredictable changes in traffic load on the QoS of WSANs. It utilizes a fuzzy logic controller inside each source sensor node to adapt sampling period to the deadline miss ratio associated with data transmission from the sensor to the actuator. This architecture is generalized to WSANs in the sense that the underlying WSN is not specified. The motivation is to provide reliable QoS for sensor-to-actuator communication for timely task allocations and situational awareness. The authors provided a good design overview for the FIS, then simulated a WSAN with their method and compared it against the same WSAN without it, hence showing that their proposed algorithm has merit.

The study in [41] is an extension of [132], where fuzzy rule extraction for the FIS is solved by means of a GA. The authors provide a simulation study and discuss the results, then compare their approach against the one in [132], thus concluding that their technique yielded substantially better results.

Oda et. al. [133] introduced a Deep Q-Network for controlling the mobility of the actor nodes in a WSAN so as to ensure their connectivity. This will help the actors respond to events in the field. The authors considered tasks in which an agent interacts with an environment. In this case, the mobile actor node moves step by step in a sequence of actions, observations and rewards. The simulations were conducted on a synthetic 3D environment and using the Rust programming language.

C. Sink Mobility

The following are relevant works for the Sink Mobility problem in WSANs. Table IV gives an overview of the surveyed works. The CI-based solutions in this area fall along two major groups: (i) optimization of the sink path without network clustering and (ii) joint network clustering and sink path optimization.

1) Sink Path Optimization: Zhong and Zhang [139] described an algorithm that does not attempt to create clusters of nodes. Instead, it tessellates the AOI into a grid, then determines an optimal mobile sink trajectory with an ACO algorithm in regards to the average residual energy in a cell, the number of communication hops to reach the sink node for all nodes, and the distance the sink node has to travel. The algorithm is validated through simulations with a comparison to two other non-CI methods.

Ho et. al. [140] approached the sink mobility problem by determining the optimal trajectory that a UAV must take to optimize travel time, energy use and communication bit-error. They leaned on the PSO optimizer for this task. The authors remarked that previous research has shown that communication errors are influenced by the UAV's angle of approach, hence they added it as one of the objective functions for the PSO algorithm. A thorough simulation and empirical analysis of their method against the LEACH-C [149] algorithm was provided, thus showing the benefits of the proposed scheme.

Cai et. al. [137] consider the case of multiple mobile sink nodes. This is then formulated as a TSP problem with multiple agents and solved via a GA that optimizes the communication energy costs. A simulation is given, and a comparison to two other methods is provided.

Instead of pre-determining an ideal route, Sarvestani et. al. [146] divided the AOI into regions, then assigned each cell a value as per an FIS. The FIS inputs are: a region's residual energy, the average number of nodes in the region and the data generation rate. The sink node then relocates to regions with higher scores, thus dynamically creating the sink node's path. A simulation is provided to validate the method.

The work of [138] starts with predefined clusters. The sink mobility problem is then reduced to a TSP with cluster

TABLE IV
APPLICATION OF CI TECHNIQUES TO THE SINK MOBILITY PROBLEM IN WSANS

CI Technique	Algorithm	References	Validation	Computation Distribution
Evolutionary Algorithm	AIS	[136]	Simulation	Centralized
	GA	[39], [137], [138]	Simulation	Centralized
Swarm Algorithm	PSO	[139]–[141]	Simulation	Centralized
	BA	[142]	Simulation	Centralized
	ABC	[143]	Simulation	Centralized
	ACO	[144]	Simulation	Centralized
	BFA	[145]	Simulation	Centralized
Fuzzy Logic	FIS	[146]	Simulation	Centralized
Learning Systems	SOM	[147]	Simulation	Centralized
	Bayesian classifier	[148]	Simulation/Real-world prototype	Centralized/Distributed

heads as waypoints, for which a GA is then chosen as the solver. The method optimizes the time needed to complete the trajectory and the energy usage of the cluster heads. A validation by simulation is given, including a comparison to a static predetermined mobile sink path.

Comarela et. al. [144] consider sink mobility in sparse networks. In such scenarios, there might not be enough nodes to either create clusters or form reliable communication paths. They then formulate the problem as a TSP-N instance and solve it by using ACO to minimize the sink's trajectory length. Following the execution of the ACO method, another ACO process is started, thus making this a continuous ACO process. A simulation is given, along with a comparison to another TSP-N solver.

Kavitha [145] put forward a BFA-based algorithm for data collection with mobile sinks. This technique initially built an optimal tour path through a TSP solver. Then, the generated Hamiltonian paths are divided into multiple loops by means of the BFA algorithm, where the fitness function takes into account the average delay of each loop. Multiple loops are formed such that the total delay is minimized. Moreover, the proposed approach combines multiple aggregation tasks for enhanced energy efficiency. Simulation results showed that the proposed technique achieves better performance in terms of packet drop, delay, delivery ratio and reduces the energy consumption by 17%.

2) Joint Network Clustering and Sink Path Optimization: Abo-Zahhad et. al. [136] proposed a system based on the AIS metaheuristic algorithm to jointly optimize the sink path along the WSAN and the number of cluster heads. The solution is then evaluated on the required communication energy and number of control messages. The method is validated by comparing it to the well-known LEACH protocol [150].

Similar to [136], the study in [39] attempts to determine an optimal sink path while creating node clusters and formulates the problem as a Traveling Salesman Problem with Neighbourhoods (TSP-N). The authors employed a single-objective GA with a fitness function based on the shortest trajectory and coupled with a clustering algorithm that creates node clusters. They proposed two new operators, namely Sequential Constructive Crossover (SCX) (to determine edges given a vertex), and Modified-SCX (to select different edges than SCX in certain cases). The proposed method is compared to a random walk algorithm.

The studies in [141]–[143] again attempt to create optimal clusters, where the cluster heads are waypoints. Saad et. al. [142] first create 1-depth clusters, then utilize BA to solve the TSP problem (i.e., finding the minimal-length path), while [141] exploits PSO to determine node clusters as well as the optimal sink path, thus optimizing the energy usage required for the cluster heads. As for [143], a cluster creation algorithm is used, then the ABC metaheuristic method solves the underlying TSP problem associated with the optimal sink path calculation, this time optimizing on the amount of collected data, trajectory length and energy expenditures. A simulation is provided for the proposed method. The authors in [143] presented an extensive validation of their work by simulations, and comparisons to a random walk trajectory and an ACO-planned trajectory.

Faigl and Hollinger [147] studied the problem of finding a cost-efficient path to collect data from a given set of sensors (not necessarily from all of them). Their problem formulation combines elements from the TSP-N and the Prize-Collecting TSP (PC-TSP) in light of the fact that the data gathered by some sensors is deemed more important than the data from other sensors. The authors applied a Self-Organizing Map (SOM), a very popular type of ANN, to simultaneously determine the sensing locations and the shortest path among them. The SOM-based method is less computationally demanding than the techniques based on combinatorial solutions to the underlying TSP.

Uddin et. al. [148] developed a dynamic network clustering algorithm for a UAV-assisted WSN responsible for crop health monitoring. The dynamic clustering of the sensor nodes takes into account several factors, including the UAV path for data collection. A Bayesian classifier is run locally at each node to determine whether it should become a clusterhead or not. The authors validated their approach through simulations and built a proof of concept with an Arduino microcontroller.

D. Topology Control

The following are relevant works addressing topology control in WSANs from several perspectives. Table V gives a brief overview of these works.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

TABLE V
APPLICATION OF CI TECHNIQUES TO THE TOPOLOGY CONTROL PROBLEM IN WSANS

CI Technique	Subproblem	Algorithm	References	Validation	Computation Distribution
Evolutionary Algorithm	Sensor Self-Relocation	MOEA	[151]–[154]	Simulation	Centralized
	Sensor Self-Relocation	MOEA	[155]	Simulation	Centralized
	Sensor Self-Relocation	AIS	[156]–[158]	Simulation	Centralized
	Sensor Self-Relocation	GA	[159]–[161]	Simulation	Centralized
	Sensor Self-Relocation	GA	[162]	Simulation	Distributed
	Sensor Self-Relocation	BBA	[163]	Simulation	Centralized
	Actuator-Assisted Sensor Relocation	MOEA	[86], [88]	Simulation	Centralized
	Actuator-Assisted Sensor Relocation	GA	[164]	Simulation	Centralized
	Actuator-Assisted Sensor Relocation	GA	[165]	Simulation	Centralized
	Actuator-Assisted Sensor Deployment	MOEA	[166]	Simulation	Centralized
	Relay Sensor Deployment	GA	[167], [168]	Simulation	Centralized
	Relay Sensor Deployment	MOEA	[169], [170]	Simulation	Centralized
	Relay Sensor Deployment	MOVNS	[171]	Simulation	Centralized
	Actuator-Assisted Sensor Replenishment	GA	[71]	Simulation	Centralized
Swarm Intelligence Algorithm	Sensor Self-Relocation	AFSA	[172]	Simulation	Centralized
	Sensor Self-Relocation	ACO	[173]	Simulation	Centralized
	Sensor Self-Relocation	GSO	[174]	Simulation	Centralized
	Sensor Self-Relocation	PSO	[175]–[180]	Simulation	Centralized
	Sensor Self-Relocation	ABC	[181]	Simulation	Centralized
	Sensor Self-Relocation	PSO	[164], [182]	Simulation	Centralized
	Actuator-Assisted Sensor Relocation	BFA	[182]	Simulation	Centralized
	Actuator-Assisted Sensor Relocation	ACO	[183], [184]	Simulation	Centralized
	Actuator-Assisted Sensor Relocation	ACO	[185]	Simulation	Localized
	Actuator-Assisted Sensor Deployment	PSO	[186]	Simulation	Centralized
Fuzzy Logic	Sensor Self-Relocation	FIS	[187]	Simulation	Distributed
Hybrid System	Sensor Self-Relocation	EA/PSO	[188]	Simulation	Centralized
	Sensor Self-Relocation	FIS/MOEA	[61], [120]	Simulation	Centralized
	Actuator-Assisted Sensor Relocation	FA/HaS	[189]	Simulation	Centralized
	Actuator-Assisted Sensor Relocation	FIS/MOEA	[87]	Simulation	Centralized

1) Sensor Relocation: The two main avenues under this category are sensor self-relocation and static sensor relocation by mobile actuators.

Node self-relocation

The approaches listed in this subsection often consider multiple aspects behind the optimization process, such as energy expenses resulting from node mobility and the amount of improved/restored coverage.

The authors of [172] propose a scheme where mobile nodes move following a random deployment in order to augment the coverage of the network. They employ a modified AFSA algorithm to discover new positions that optimize the network coverage. Their method is then simulated and compared against an unmodified AFSA, thus demonstrating that it results in higher network coverage.

Network reconfiguration with mobile nodes for optimal AOI coverage is considered in [174] by using a modified Glowworm Swarm Optimization (GSO). This algorithm optimizes the distance traveled by the nodes, the energy use of the reconfiguration, and the redundant coverage of the solutions. A simulation is presented that compares the GSO-based method to another technique. This comparison is done for different network densities and shows that the proposed method yields superior results.

Ni et. al. [175] put forth another coverage optimization method for mobile nodes based on a modified PSO. The PSO determines appropriate positions for optimal coverage and minimal moving distance. It is argued that Quality of Service (QoS) is related to coverage, hence QoS is also taken into account during the optimization. Finally, the method is simulated and compared against a basic PSO and another PSO-based coverage algorithm, where the proposed method performed best.

Another algorithm leaning on PSO to increase the network coverage is brought forth in [177], [188]. The goal is to redeploy a set of mobile robots according to the network node density for repairing sensing coverage holes after the initially random sensor deployment. The method in [188] employs a PSO with GA principles such as selection and mutation to fix some PSO problems whereas [177] proposes an improved PSO method. Both techniques are validated via simulation analyses.

In [151], a MOEA is used to search for optimal node positions that lead to increased coverage while minimizing the average distance traveled by the actuators. The method is validated by simulation.

A modified PSO is again unveiled in [178], with a single decision objective related to network coverage. Virtual forces are first created to guide the PSO algorithm. Then, multiple PSO algorithms are run in parallel, with the best solutions influencing other iterations in an algorithm called co-evolutionary PSO. The method is validated by experimentation.

Salehizadeh et. al. [179] designed a new PSO implementation, the Individual PSO [190], for the network coverage problem. They argued that PSO is not fast enough to be used in real-time scenarios while their Individual PSO algorithm is. The algorithm reduces the population of the PSO algorithm

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

to one individual and adds a chaotic coefficient based on Logistic Map [191] to make up for the lack of group search behaviour which forms the basis of PSO. The algorithm aims to maximize coverage by relocating mobile nodes, where they use a probabilistic detection model. A target has a probability of detection in such a model, contrary to the binary detection model, where an object will necessarily be detected if it is within the sensor's coverage. Their method is then simulated and shown to outperform the classical PSO for the network coverage maximization problem.

In [61], [120], nodes self-bid for their suitability to partake in a network self-healing task after a node failure due to e.g., battery depletion. The proposed approach is governed by market-based task allocation principles where network nodes make use of an FIS to compute their bids. Then, a MOEA is used to determine optimal positions to augment coverage. The FIS takes as inputs the node's distance to the event's location (failed node's position), the node's battery level, and a function of its coverage at its current location. The MOEA then optimizes both restored coverage and energy usage. A simulation is given where one [120] and multiple [61] areas of reduced coverage are identified and the method proves effective at having the network repair itself.

The authors utilized a single-objective AIS [157], then a multiobjective AIS [156] scheme to repair coverage holes. In [157], the algorithm maximized a fitness function based on energy usage and restored coverage, while in [156], these are treated as separate objectives. Both are compared against a similar coverage-restoring algorithm based on a MOEA that shows that the proposed algorithm performs better.

Katsuma et. al. [159] introduced a GA-based algorithm that optimizes a fitness function based on energy expenses and network coverage. The proposed method places a special focus on communication problems by providing solutions that can provide a connected tree to the sink node and k-node coverage over certain areas. The method is experimentally validated.

In [95], a MOEA-based approach that maximizes coverage and minimizes future energy expenses for communication is considered. The algorithm determines ideal positions for node relocation as well as clusters following the relocation. The method is validated by simulation and compared to existing algorithms.

In [181], the authors developed an ABC algorithm to increase coverage using mobile nodes with no other objective function. The method is validated by simulation.

Jiang et. al. [162] elaborated on a distributed way to increase coverage using a GA. Each node first determines where it could move. Then a solution is evolved as per the node's local information. The node then moves to that location, and this process is repeated. The method is validated by experimentation.

FL is used in [187] to determine the extent to which a network node should move in order to improve the overall coverage. Nodes input the distance to their neighbourhood as well as the distance to borders and obstacles into an FIS. The FIS then outputs a new position that slowly spreads the network by repelling nodes from each other, hence increasing the total coverage. The method was validated by a simple simulation against other common approaches.

A MOEA-based approach that optimizes network lifetime, moving costs, and coverage, while being constrained on communication success rate is proposed in [153]. This method is then validated by simulation.

While many authors only consider the binary sensing model as the underlying sensing scheme, the authors of [154] considered a probabilistic sensing model and generated probabilistic coverage from it. They then tried to determine optimal coverage in this probabilistic model and aimed at minimizing movement distance by comparing 2 MOEAs and 2 multi-objective PSO algorithms. The method is thoroughly simulated under several network densities with all four methods. The experimental analysis confirms that MOEA/D [192] performs better than all other optimization methods.

In [176], a coverage maximization method based on PSO and Voronoi cells is investigated. First, Voronoi cells are computed from the static sensor nodes. Then, a PSO algorithm that attempts to increase coverage makes use of these cells to reduce redundant coverage by deploying mobile nodes in optimal positions. The method is validated by simulation. Similarly, the approach in [160] also computes Voronoi cells but utilizes a single-objective GA to determine the nodes' optimal positions. The work in [176] is also validated by simulations and the effect of the number of static nodes upon the network coverage is studied.

Banimelhem et. al. [161] brought forward a GA-based optimizer to determine the ideal number of nodes to deploy in order to repair all network holes. The method is then validated by simulation.

The authors of [180] employed the Quantum PSO method to determine actuator placement in WSANs for optimal sensor node coverage. This method enhances the exploration ability of the normal PSO. They devised a fitness function that evaluates the distance from actuators to sensors and the number of actuators used, with the intention to minimize it. Their method is validated by a brief simulation study.

The work in [163] instead approached the coverage problem by using a single-objective metaheuristic algorithm, Biogeography-Based Optimization (BBO) [193]. The algorithm works by simulating the migration of species in search of better habitats. The method is simulated and compared against ABC, Stud Genetic Algorithm (SGA) [194], and a method based on PSO, the later version of [178]. It is shown that the BBO-based method outperforms its peers.

Kuang and Cai [158] took a different approach. They first computed optimal positions for coverage maximization. Then they resorted to an AIS to assign these positions to nodes accordingly, hence minimizing the distance moved. This can be seen as a two-step optimization, first on coverage, then on moving distance. The algorithm was simulated and shown to beat an existing GA-based approach.

Yu et. al. [155] considered a hybrid sensor network (i.e., a network containing both static and mobile units) where the mobile nodes self-relocate to heal coverage holes in the network. The number of coverage holes and their size are first detected by using a level set method. A bi-objective MOEA-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

based optimizer then decides where to dispatch the mobile nodes in order to restore as much coverage on average as possible while minimizing the average distance traveled by the mobile nodes.

Actuator-assisted sensor relocation

Robot-assisted sensor relocation (RASR) is a challenging optimization problem that emerged in WSRNs [13] [63]. Falcon et. al. [183] envisioned a mobile robot replacing damaged sensors with spare ones (passive sensors) gathered from the field. The optimal sensor relocation trajectory followed by the robot that departs from and returns to the base station is NP-hard to compute. The authors resorted to ACO algorithms, in particular the Max-Min Ant System implementation, to compute a high-quality suboptimal trajectory in a short time. They studied the impact of six heuristic functions on the ACO method and showed that it outperformed Simulated Annealing. Later on, Mou and Dai [184] reported superior results over [183] with another ACO approach: an Ant Colony System using constrained neighborhood search and a special mutation operator. A GA-based scheme that outperformed [183] was vaguely described by Shams and Khan in [165].

The multi-robot case was analyzed in [189]. The problem was formulated as a special case of the Vehicle Routing Problem with Selective Pickups and Deliveries. A hybrid meta-heuristic of Firefly Algorithm (FA) and Harmony Search (HaS) was designed to calculate minimum-length sensor relocation paths for all robots.

The above works only cared about optimizing the length of the relocation path(s) to be followed by the mobile robot(s). Desjardins et. al. [86] additionally considered the quality of the relocated sensors as this aspect will impact the network lifetime. A MOEA-based solution that minimizes the trajectory length, maximizes the chance of reducing coverage holes and maximizes the time for which the coverage is utilized was put forward in this study. The method includes an operator that repairs solutions with numerous damaged sensors. A variety of MOEAs were tested in the simulations. This method was further extended in [87] by incorporating a risk management component, where nodes that are more at risk of being damaged are flagged by an FIS as such. Two more MOEAs are added to the simulation experiments. In [88], the authors studied the multi-robot reliable sensor relocation scenario. A fourth optimization objective was added to measure load balancing, and the algorithm was reworked to consider all the robots. This new technique was simulated with the same MOEA algorithms reported in [86].

Wang et. al. [185] proposed an ACO-inspired localized algorithm to help a team of robots relocate sensors and improve their area coverage. This algorithm considered relocating not only spare sensors but also active sensors whose area coverage is mostly overlapped by neighboring nodes. Each robot may carry at most one sensor and calculates pickup/drop-off probabilities based on locally detected information. Two variants of the localized scheme are put forth: one optimizing the total area coverage and the other one optimizing the cost of robot movement.

2) Sensor Deployment:

Static sensor deployment

Singh and Kumar [164] designed an algorithm for the deployment of static sensors via an Unmanned Aerial Vehicle (UAV). This UAV first takes pictures of the AOI, then segments it. The authors relied upon a GA, then a PSO method to identify optimal positions for coverage from the segmented image by optimizing on coverage only, with the UAV accomplishing the task of relocating the nodes. Both CI techniques are used and compared with the GA-based one producing superior results.

Similarly, the authors of [182] described a method for sensor deployment through a UAV. The UAV takes images of the AOI and then segments them. PSO and BFA are then used to optimize this segmentation that ultimately yields efficient sensor deployment positions. The method is then validated by simulation.

Saadallah [166] studied the scenario where mobile robots deploy static sensors at pre-computed locations that guarantee optimal network coverage and connectivity. She leaned on NSGA-II as a multiobjective optimizer in order to minimize the deployment latency while achieving good load balancing among the robots. NSGA-II was afterwards seeded with the 2-opt heuristic so as to improve its convergence and attain more robust solutions.

Cheng et. al. [186] envisioned a mobile robot deploying additional sensors to maintain the network coverage in presence of node failures. Depending on their capacities (e.g., higher energy reserves), these extra sensors are sometimes referred to as *relay nodes*. A PSO implementation with a linearly decreasing inertia weight optimized the network's maintenance cost, which consists of three factors: coverage rate, node residual energy and node consumption energy. This method achieved relatively higher coverage rate and a much longer maintenance period than random and uniform redeployment algorithms.

Relay node deployment

Unconstrained relay node deployment

Optimally deploying relay nodes with a mobile robot entails NP-hard complexity. Lanza-Gutierrez and Gomez-Pulido [171] investigated the use of two Multi-Objective Variable Neighbourhood Search (MOVNS) algorithms to deploy relay nodes in a single-tiered WSN with the goal of optimizing the average energy consumption and the average sensitivity area of the network. Peiravi et. al. [167] put forward a clustering method powered by a GA in homogeneous two-tiered WSN; their goal was to optimize the network lifetime with different delay values. Azharuddin and Jana [168] aimed to minimize the number of relay nodes and maximize network connectivity by means of a GA in two-tiered WSNs. Perez et. al. [169] employed a MOEA to optimize both the energy cost and the number of routers in a single-tiered WSN. None of these approaches imposes any constraint on the location of a relay node in the monitoring region.

Constrained relay node deployment

The *Constrained Relay Node Deployment Problem* (CRNDP) was solved in [170] via three well-known multiobjective optimizers, viz NSGA-II, AbYSS (based on Scatter Search) and MOPSO. They aimed at minimizing the average energy consumption of the sensors while maximizing the average network reliability. The performance of the three algorithms was gauged in terms of hypervolume and coverage of two sets. They concluded that NSGA-II is the best performing technique followed by AbYSS and then MOPSO.

3) Sensor Replenishment: Ye and Wang [71] designed a GA-based solution for the calculation of a mobile robot's trajectory to replenish the energy source of multiple static sensors in the WSAN. The problem is very similar to the TSP with time windows (TSP-TW), for every static node must be served within a certain time period (i.e., before it runs out of energy). The authors created custom genetic operators and a local search strategy for this problem. They validated their approach via simulations on a 14-node network but did not compare their results against those of any TSP-TW solver.

A similar situation is found in [173], but ACO is used instead. The nodes' energy is used as the pheromones, with robots replenishing sensors on their paths. A simulation is presented but no comparison with other methods is included.

E. Localization

The following are relevant works on the localization problem in WSANs (both range-based and range-free). CI techniques are used to either directly infer possible locations or to refine/optimize their estimates. Table VI presents a brief overview of the surveyed works.

1) Range-Based Localization: Herrero and Martínez [201] tackle localization as a fuzzy estimation problem, where the position of the node and the sensor measurements can be described by fuzzy sets. Fuzzy densities are then used over all possible locations of a node, which are the vertexes of a Voronoi-tessellated environment. These fuzzy densities are then increased or decreased upon receiving packets from anchor nodes by using RSS. This method explicitly focuses on locating mobile nodes in WSANs in indoor spaces; the authors chose FL to deal with the information uncertainty. The proposed technique was tested in a real scenario with mobile nodes and outperformed a Monte Carlo localization approach.

Similarly, the authors of [203], [204] harness metadata from communications to determine a proper location for a mobile node moving through a WSAN. Gholami et. al. [203] leaned on the ToA from a particular node to each anchor node as the input to an ANN, which then has two outputs: the X and Y node coordinates in a two-dimensional localization problem. Irfan et. al. [204], a predecessor of [203], employed RSSI and Link Quality Indicator (LQI) as inputs to the ANN. LQI denotes the quality of the link, i.e. its error rate. In [203], the method is implemented on the sink node, though they note that this scheme could also be run locally on each node. A one-hidden-layer ANN, then a two-hidden-layer ANN, are exploited for validation purposes in [203] by simulating and comparing their approach to a trilateration-based localization method. The empirical evidence indicates that the proposed localization method performs better than the benchmark trilateration technique. The algorithm in [204] is compared to other existing localization methods, with [204] outperforming them all.

Chan and Wen [78] again used FL and a range-based method for localizing mobile nodes in WSANs. They relied on the ToA to determine an approximate position, which is refined afterwards with AoA data to train an FIS. This FIS can then adjust the estimated position on the (X,Y) plane to give a refined estimate. Their proposed method is simulated along with two other methods, and the results indicate that the proposed algorithm outdoes them both.

In [195], a node first estimates its position by using a weighted centroid method fed by messages from a mobile anchor node. The positions transmitted by the anchor nodes are weighted in such a way that positions closer to the node have a higher weight, with the distance to the node being determined by RSSI. This estimate is then refined using a GA. The proposed method is simulated and compared against another localization method.

Karedla and Anuradha [196] put forth a two-step procedure to localize mobile nodes. First, an estimate is calculated using the weighted centroid algorithm. Then, a GA is responsible for refining the position estimate by making use of the difference between the estimated positions and the actual RSSI-inferred distances to the anchor nodes as a fitness function. Their proposed method is verified by simulations and compared to two other algorithms.

Kulkarni and Venayagamoorthy [182] introduced an algorithm for iterative localization in dynamically deployed WSNs. The network contains three types of nodes: simple sensor nodes, anchor nodes, and a UAV that deploys these. When a node contains three or more localized nodes inside its communication range, it will determine its distance to them, create an estimated position for itself, then use either BFA or PSO to optimize this estimate. Whenever new nodes get deployed, the previously localized nodes can themselves become anchors, thus iteratively localizing the network. This method with both CI techniques is validated by simulation but is not compared to other algorithms.

Guo and Tang [205] brought forward a localization method for mobile nodes in WSANs. An SVM for each dimension is first trained by using estimated distances to the other anchor nodes in the respective dimensions, where a class is an interval in the dimension space. They then used the trained SVMs to help localize new mobile nodes, thus refining the estimate through predictions of the next location based on simple movement dynamics. Finally, a filtering step takes place to remove possible invalid locations with the help of messages from the anchor nodes. The proposed method is validated through a simulation analysis and compared to two other schemes.

2) Range-Free Localization: Bao et. al. [199] attempt to localize nodes in WSNs with mobile anchors via a range-free mechanism. The mobile anchors periodically broadcast their locations. Upon receiving such a message, lost nodes will retain it for future use and re-transmit it once to its 1-hop neighbours. When enough of these messages are received, a PSO algorithm kicks in, where each particle represents a possible position for the node. This algorithm converges to a position denoting the node's best guess as to where it is located. The proposed scheme is simulated and compared

TABLE VI
APPLICATION OF CI TECHNIQUES TO THE LOCALIZATION PROBLEM IN WSANS

CI Technique	Subproblem	Algorithm	Reference	Validation	Computation Distribution
Evolutionary Algorithm	Range-based	GA	[195], [196]	Simulation	Centralized
	Range-free	GA	[197]	Simulation	Centralized
	Range-free	DE	[198]	Simulation	Centralized
Swarm Intelligence Algorithm	Range-free	PSO	[199]	Simulation	Centralized
	Range-based	PSO, BFA	[182]	Simulation	Centralized
	Range-free	PSO	[197]	Simulation	Centralized
	Range-free	ACO	[198]	Simulation	Centralized
	Range-free	ACO	[200]	Simulation	Centralized
	Range-free	Cuckoo Search	[40]	Simulation	Centralized
Fuzzy Logic	Range-based	FL	[201]	Simulation	Distributed
	Range-based	FIS	[78]	Simulation	Distributed
	Range-free	FIS	[202]	Simulation	Distributed
Learning Systems	Range-based	ANN	[203], [204]	Simulation	Distributed
	Range-based	SVM	[205]	Simulation	Distributed
Hybrid System	Range-free	ABC/GA	[206]	Simulation	Centralized

against a centroid-only variant.

The studies in [40] [198] [197] solved the localization problem by means of biologically inspired optimization methods: CS, ACO, DE, GA, a GA-Simulated Annealing technique, and PSO. They first prepared a set of estimated node positions by having a mobile anchor node periodically broadcast its location and triangulating the nodes' locations. The first and the last broadcast positions received by a node are assumed to correspond to the communication range of this node. Multiple possible locations can be eliminated with the help of neighbours, though this is not done in [198], where the multiple locations serve as a basis for the optimization algorithm. This set of estimates is then refined using one of the aforementioned optimization techniques. They simulate their proposed methods and benchmark them against the unrefined estimation algorithms, thus showing encouraging results.

The authors of [200] approached the localization problem from another perspective. Instead of optimizing position estimates, they proposed a method that determines ideal broadcast locations that would give accurate triangulation positions. This can then be formulated as a TSP for a mobile anchor node. This problem is then solved by using ACO to determine an ideal route. The proposed method is validated by simulation and compared to a static route.

Similarly, Qi et. al. [206] formulated the localization problem as a TSP instance. The proposed localization method is simple: the mobile anchor node essentially visits each node that requires localization by passing within a distance threshold of it. Then, it localizes the node by using its own known position. The authors solve the TSP problem via a hybrid ABC-GA algorithm. The proposed method is validated by simulation.

Dutta et. al. came up with a coarse-grained localization technique in [202]. The authors described a system where the AOI has been tessellated into a grid and a mobile node can be localized to one of the grid cells by using a FIS. First, the FIS is trained with past data to recognize patterns in the node's movement. This trained FIS can then indicate which cell the mobile node is presently in. The authors describe and define their system, then validate it through simulations.

F. Summary

The WSAN *Actuation* problem category has been tackled from many standpoints using CI techniques. Those approaches that revolve around Task Allocation often resort to market-based allocation techniques optimized via EAs/SIAs to satisfy the overall system goal. There is plenty of room for the application of MOO methods in this area. Another popular trend is to employ FIS/ANN to design control systems for these actuators that allow them to individually bid for certain tasks. Regarding the subset of Actuation approaches concerned with task execution via actuator coordination and event prediction, FIS and FL are the main CI schemes employed to ensure a smooth coordination among the actuators, although we see an emerging interest in RL and MDP as LS representatives. The optimization angle is still present via EAs/SIAs solving different manifestations of actuator coordination problems such as target tracking or path planning. A vast majority of the proposed approaches rely on a centralized computation architecture.

In the WSAN *Communication* category, the application of CI techniques to the routing subproblem is confined to solving optimization problems primarily via SIAs. The communication routes are mainly static (i.e., do not change over time) except [130] that envisions dynamic communication backbones. Multiple aspects of these routes such as energy consumption, signal strength or message latency are taken into account during the optimization process. In the clustering subproblem, the suitability of a WSAN node to become a cluster head is modeled through an FIS and the selection of potential cluster heads network-wise is entrusted to EA-based optimizers. Finally, the QoS sub-problem is the least explored by CI techniques. The few available works are related to fuzzy control and genetic optimization at the node level to ensure reliable sensor-actuator communication.

Concerning the *Sink Mobility* category, CI optimization techniques have the upper hand as they try to derive the best path for the mobile sink. Some studies simultaneously identify the most suitable cluster heads in the WSAN. A few works depart from the traditional problem formulation by considering special cases such as multiple mobile sinks or a sparse network. Finally, an FIS to gauge the attractiveness of the network regions for sink visitation was also put forth. MOO methods as well as LS/HS schemes would be a great addition to the repertoire of CI applications here.

The CI presence in the WSAN *Topology Control* category is largely dominated by EA/SIA-based optimization methods across all its subproblems, namely sensor deployment, relocation and replenishment. This is quite understandable since modifying the WSAN topology serves an ultimate goal, e.g., maximizing network lifetime or expanding/restoring network coverage. We do see increasing evidence of the successful synergy between FIS/DL and nature-inspired CI optimizers in the sensor relocation arena. Sensor replenishment by mobile actuators is an exciting and largely uncharted territory for new CI applications.

Finally, in the WSAN *Localization* problem, we notice that range-based methods have been slightly more studied through CI techniques than their range-free counterparts. The need to reason under imprecise information (coming from unreliable distance estimates of the nodes) makes it an appealing choice for the application of FL/FIS and LS (ANN/SVM) techniques, with some genetic and swarm-inspired optimizers in the background to produce an accurate solution. The latter category (range-free localization methods) hinges more heavily on EA/SIA-based optimization given the rather reliable estimates of the anchor nodes' position that are broadcast to the rest of the WSAN.

VII. Discussion

This section discusses which CI techniques are suitable to solve each WSAN problem, and whether they were found in the surveyed works. Figure 14 unveils four of the problems together with their sub-problems, the most suitable CI techniques, and the number of papers found in that category. The actuation problem is not included since it is more of a overarching process and can not be neatly separated from the other problems. Furthermore, the most appropriate CI technique is highly dependent on the approach taken to solve the problem. More specifically, we elaborate on the main findings drawn from Section VI along each type of WSAN problem.

A. Actuation

The control process in WSAN is complex, with examples ranging from controllers in sinks for simple actuation tasks to complex coordination for exploration, foraging, or tracking. Additionally, each paper touched on one step of the process, while others described a more comprehensive behavioral control for the WSAN. As such, it is difficult to categorize each one of them and compare them. It can be seen however that many CI techniques are used to resolve each of the steps either in combination or separately.

The task creation step is ripe with ambiguity and uncertainty, and is closely related to data fusion and event prediction. Consequently, LS and FL should increase their presence here. If we look at the works, it can be seen that they are in fact exploited in many works that concern task creation, such as [121] and [77], or for prediction such as in [118], [119]. Planning appropriate tasks is key for search or exploration operations, and it can be noticed that the relevant works with those goals could also employ techniques such as those in [93], [112]. In fact, [82] presents one method based on FL and another on SIAs, the two dominating CI techniques used.

Actuator selection is also often conducted to provide a better set of nodes for task allocation or to reduce the complexity of the task allocation process, such as in [62], [105]–[107]. However, task creation, actuator selection, and task allocation are often put together for greater synergy. All of these are highly dependant on each other, and researchers exploit this dependency by using EAs to directly discover the best combination of created tasks, actuators, and allocation. This can be witnessed in [62], [91]. The MRTA problem is known to be a combinatorial optimization problem [207] so this is not surprising.

As for coordination, it is highly dependent on the task to solve. For example, ACO is used in [90] for optimal and cooperative path planning, while FL is used in [103] for a mobile node to position itself. Another example is the conflicting behaviour resolution via an FIS found in [113]. Coordination requires fine cooperation and communication among mobile nodes, hence implying unreliability, and may be why FL is found more commonly in this area.

B. Communication

The routing problem must evaluate many candidate routes in an attempt to find the best one according to some criteria, hence EAs and SIAs best fit this problem. However, all network node data may not be available to a given node at all times. Previously used routes may have had greater success, an idea that makes Machine Learning relevant here. Node clustering could be cast as a combinatorial optimization problem, where a subset of the nodes in a network must be chosen per some constraints and predefined evaluation functions, and consequently is solvable with EAs and SIAs. Since communication in WSANs is fundamentally unreliable, information accuracy and availability must be taken into consideration. Consequently, FL and ANNs may be best suited in this area.

In the routing case, most works used pure SIAs, with one resorting to a hybrid algorithm in order to circumvent some of the issues in SIAs. This was expected due to the reasons explained above. However, none of them used route learning [31], [208] and that fact leaves some research potential in this area. As for the combinatorial optimization problem of clustering, the most represented CI technique was GAs, with some FL to determine the suitability of a node to become a cluster-head. Finally, the QoS problem has not been fully explored from a CI-WSAN perspective. Only one branch of work for QoS in WSAN has been explored. More complex QoS solutions, such as data transmission scheduling, have not been addressed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

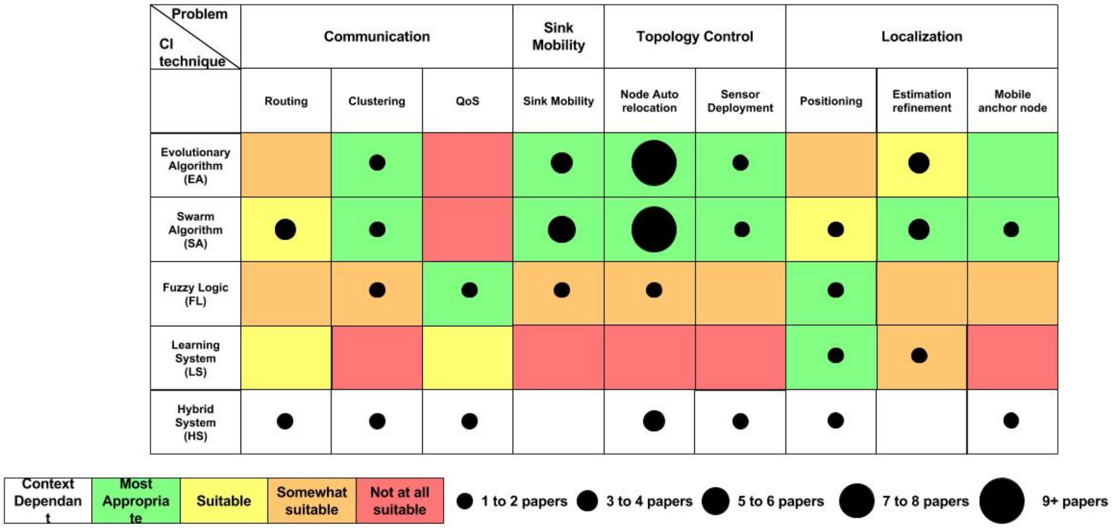


Fig. 14. WSAN problems, suitability of CI techniques and their representation in the surveyed works

C. Sink Mobility

The traditional sink mobility scenario (i.e., mobile sink, static sensors) is often formulated as a TSP instance, since it is essentially a combinatorial optimization problem where the sink must choose the best path to visit all static sensors as per the underlying optimization criteria. This classical scenario is also formulated as a TSP-N instance since the sink node does not necessarily have to physically reach the static sensor nodes, for it must only travel inside a node's communication radius. Furthermore, it might not be efficient to visit every node. Many approaches in fact generate communication clusters before determining an optimal trajectory. This complicates the problem by needing to create the most efficient cluster configurations that can yield the best path for the sink to travel along. The CI techniques most suitable for solving combinatorial optimization problems fall under the EAs and SIAs categories. The relevant works surveyed confirm this point, since all but one paper utilized either an EA or an SIA.

The sink mobility scenario where sensor nodes are also mobile has not been considered in the surveyed works yet. This problem formulation no longer follows the TSP assumptions, hence another model has to be designed. We envision dynamic optimization via EAs and SIAs as a fitting mechanism to address this challenge.

D. Topology Control

While both coverage problems are unique, they can both be shown to be essentially combinatorial or numerical optimization problems. For sensor deployment/relocation, identifying damaged sensors and replacing them with new sensors is essentially a function mapping elements of the operational/active node set to the unused/passive set. Determining an ideal route that visits every element of the unused set can be solved by any optimization method so long as the right solution encoding and search operators are brought into effect. This problem can be further complicated by adding a subset of the operational node set to that of the unused set, with the goal to relocate those operational sets. This problem is in fact a TSP variation known as the TSP with Selective Pickup and Deliveries.

This formulation itself is compatible and directly relatable to the nature of EAs and SIAs as nature-inspired optimizers. For example, EAs consist of a population of individuals, also called chromosomes, each containing one or more values (genes). A fitness function maps the chromosome to a fitness value that the EA aims to optimize. Using this value as the key of a chromosome, and positions of nodes as genes, a solution to the Sensor Deployment problem can be found. The EA would simply explore some of the possible combinations, ordering them, and keeping the best found yet. For coverage, a fitness function must include coverage evaluation. Consequently, it is expected that EAs and SIAs dominate in this area.

Similarly, the coverage problem approached from the node auto-relocation angle can also be shown to be a combinatorial optimization problem. The n-dimensional bounded spaces contain infinite point possibilities. Luckily, digital computers do not work with such continuous spaces, but with discrete spaces that naturally convert the space into an n-dimensional grid, with a finite set of cells. The problem then becomes that of relocating the nodes to these cells in such a way as to improve coverage without increasing the number of nodes, or increasing it slightly. Note that this can also be applicable to the previous Sensor Deployment problem. This is fundamentally a combinatorial optimization problem, that of finding the best combination of points as defined by a fitness function. As previously seen, EAs and SIAs, especially PSO, are ideal CI methods to tackle these problems.

A numerical optimization version of this coverage problem can also be envisioned assuming that we are interested in the real (x,y) coordinates of the nodes in the 2D plane.

As expected, it can be seen that EAs are used for both

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

Node Deployment works, while EAs and SIAs represent the overwhelming majority of CI techniques used in Node auto-relocation. Many works also combine these categories to create hybrid systems that aim to reduce/compensate for the flaws of the other scheme.

Sensor replenishment by mobile actuators in WRSNs is largely uncharted territory for the application of CI-based optimizers, viz EAs and SIAs. The majority of the works that solve the underlying NP-hard problem of deriving the replenishing cycle do so by leaning on greedy heuristics and, to a lesser extent, on exact methods that can only handle small or medium-sized problem instances.

E. Localization

There are many perspectives on the localization problem. In general, there seems to be three major localization approaches that resort to CI methods.

The first one uses CI to directly determine a node's position, such as in [78], [199], [201]–[204]. The second approaches the problem by first creating an estimate of the node's location and then refining it [40], [182], [195]–[198], [205]. Finally, an attempt is made to optimize the trajectory of a mobile node over appropriate locations for positional broadcasts [200], [206].

The first method attempts to find positions directly without estimates, therefore making CI techniques that can deal with uncertainty more appealing, which explains the more common uses of FL and LS. The second method searches the bounded solution space given by the initial estimation, and makes EA and SIA suitable to this kind of problem. Finally, the third perspective formulates this problem as a TSP instance, and consequently, EAs and SIAs are chosen as the fittest solvers.

Within the related works, FL and LS arise as the predominant CI techniques, as they attempt to reason with uncertainty in inherently unreliable communication environments. EAs and SIAs are largely utilized in the second category, since they attempt to optimize the location estimates by reducing the inconsistency between the estimated position and the available information. Finally, in the third category, SIAs and EAs also dominate as all of them formulate it as essentially TSP instances.

VIII. Future Research Trends

In this section we first review some statistics that point to the status quo of the CI-WSAN interplay and discuss future research trends along each kind of WSAN problem studied in this survey.

A. The Status Quo

Figure 15 gives the distribution of each type of CI technique applied to WSAN problems. The reader may notice that EAs and SIAs are the two most representative categories of CI algorithms across the five types of WSAN problems discussed in this survey, which makes perfect sense owing to the number of challenging optimization problems WSANs have brought to light. This is particularly true for topology control problems. FL is least explored for sink mobility purposes but more heavily used within the actuation category, mainly for actor selection. The same can be stated about HSs. As this distribution reveals, the potential behind LSs has been generally untapped across the communication, sink mobility or topology control categories.

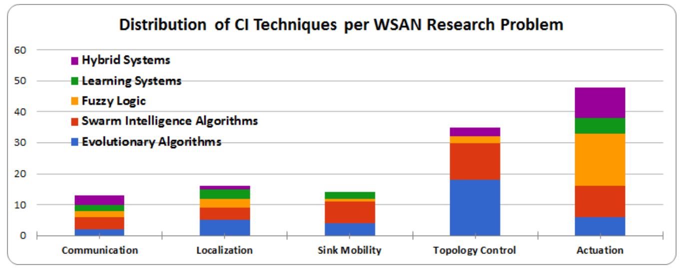


Fig. 15. Distribution of CI techniques per WSAN research problem in the surveyed works

In general, the number of research contributions featuring CI techniques to solve the WSAN problems considered in this survey has gained momentum over time. Figure 16 portrays the frequency of published CI-based works for each WSAN problem per year. While this survey has concentrated on more recent works, the figure represents the recent research trends. It can be noticed that the CI-directed research efforts into each WSAN problem are not uniform; there was a spike in interest around 2009 for actuation, followed by a slight dip, then evidence of renewed interest in 2015. Other problems such as sink mobility seem to be increasingly important in proportion.

The above findings are not surprising. Since actuation is the first problem to be added by definition when actuators were brought into the core of WSNs, it is logical that most research studies initially focused on this area. Then, as solutions for the actuation problem were found, other problems began to be investigated. In the future, this trend is expected to continue, with a growing use of CI techniques to increase the efficiency and robustness of the proposed solutions.

Another interesting graph is the CI technique distribution per year across all WSAN problems, as revealed in Figure 17. It is clearly shown that SIAs represent the dominant technique within the CI umbrella, with EAs perhaps as numerously represented. This is a clear consequence of the number of challenging and interesting optimization problems emerging within the WSAN arena.

LSs are not very well represented though this is expected to change as research into this type of technique applied to WSANs intensifies. FL is also present throughout the years. As these techniques are mastered, their particular shortcomings may be fixed by integrating concepts from other methods, thus resulting in the development of HS. These are available mostly in 2009 and 2015. HSs are expected to be increasingly important. Since combining different methods is in itself not well researched, it is normal that HSs take a more prominent role in WSANs as the research community identifies successful mechanisms for achieving powerful synergies among their constituent algorithms.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

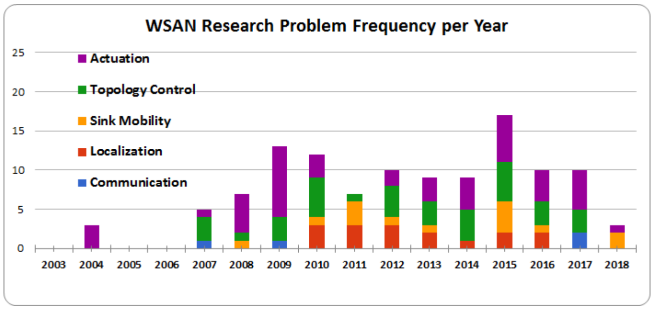


Fig. 16. Yearly distribution of WSAN research problems solved via CI techniques in the surveyed works

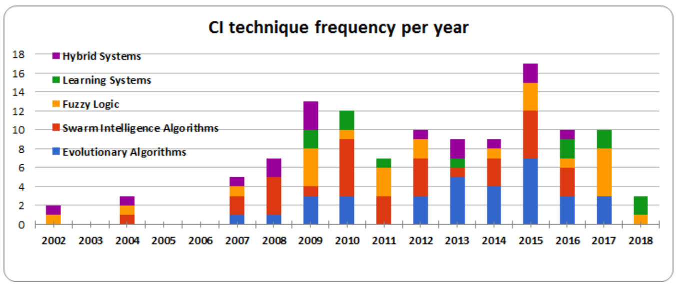


Fig. 17. Yearly distribution of the CI techniques applied to WSAN problems

The rest of this section elaborates on the authors' forecasts and thoughts on the future WSAN problems and the attempts to resolve them. New research directions as well as underutilized CI techniques are identified.

B. Actuation

The control process will only become more convoluted as WSANs start to employ more advanced CI techniques. It has already been shown that it is sometimes more efficient to combine some of these steps into a wholesome approach. In effect, CI techniques that can consider the entire process and the intricate relationships among its building blocks might perform better overall.

Another future trend is the use of systems which can handle the uncertainty of information, the probability of achieving a task, and learning the best task creation and allocation strategies. While these concepts are encountered in some of the works, they most certainly represent the future of actuation control loops in WSANs. For example, a network that knows that an event requiring resolution via actuator is likely to happen in a certain region of the AOI should be considered during task allocation. This implies event prediction via advanced LS.

Finally, none of the surveyed works take into consideration the continuous pouring in of relevant information. Once tasks are created and allocated, nodes complete them regardless of any new information that might change the task assignment outcome. These task updating mechanisms are absent and should be researched within a more robust task allocation system that leans on EAs/SIAs for dynamic optimization purposes.

C. Communication

As in other presented problems, the works considered here showcase WSANs as a recent departure from WSNs, and make assumptions that networks will be somewhat closer to WSNs. For example, routing in multi-UAV networks has been explored once. Additionally, many other assumptions are made. Network information is not always known to determine an optimal route, and consequently Machine Learning or other type of techniques may be more suitable in some WSANs. Finally, CI techniques are increasingly used, but the algorithmic methods remain simpler compared to other problems, thus leaving room for more general and optimized algorithms.

Recent work has started to explore using Game Theory or Reinforcement Learning elements in communication networks [209], [210]. The Nash equilibrium has been shown to be solvable by means of CI techniques [211]. The concept of autonomous agent fits naturally within WSNs, and even more into WSANs. Consequently, exploring how to leverage these concepts for WSAN communications from a CI perspective is a promising research avenue.

D. Sink Mobility

As networks become more dynamic, scale in size to higher orders of magnitude and head towards heterogeneity, planning a combination of static path and static clusters as seen in the surveyed papers might not work too well. The mobility of the sensor nodes themselves must also be considered when the sink nodes determine the best path to follow.

In consequence, the future of the sink mobility problem in WSANs may in fact be closer to optimization approaches that dynamically react to the network status. In this context, we envision dynamic optimization methods as valuable tools to conduct this kind of approximate reasoning. These methods will likely be driven by nature-inspired metaheuristic algorithms such as EAs and SIAs as discussed in Sections III-A and III-B. Of course, their solutions may not be optimal but will show good-enough quality and high computational efficiency. A balance between static planning and dynamic optimization will have to be taken into consideration when designing the sink mobility algorithms of the future.

Moreover, there is plenty of room for the application of FL and FISs to manage the uncertainty of several elements that influence the calculation of the optimal sink path, such as the reported sensor locations, presence of obstacles, terrain elevation and other relevant features.

E. Topology Control

The problem of Topology Control has not evolved considerably during these works. All authors roughly agree on the definition of coverage and nearly all surveyed works involves either an EA or an SIA. However, all methods are reactive to coverage failures, with some hints of proactiveness in the second method of [87]. As a result, these methods start with the premise that coverage has been impacted and must be restored. Yet, if networks would be able to predict which nodes are going to fail, they could reconfigure themselves before their actual coverage is degraded. Prediction is one of the inherent problems behind LSs and as such it is expected that more learning systems should be integrated into the LS family. Additionally, predictions usually imply probabilities, and probabilities imply uncertainty, so the use of FL and FIS as seen in [87] can also be expected.

As WSANs grow in complexity, so will the numerical/combinatorial optimization problems emanating from topology control, with more constraints and objectives. For example, the published studies generally do not consider the effects that topology control will have on communication, the future lifespan of the network, or even varying the k-redundancy over the AOI. Better multi-objective optimization methods must be researched. The authors of [179] explore this topic, mentioning that PSOs are slow and can not be used to achieve a quick resolution. It is thus expected that more advanced MOO methods that bring together several ways to improve convergence will be designed and applied for topology control in WSANs. LSs may be also employed to characterize different topological conditions in the WSAN and derive optimal policies via RL or ANNs to appropriately respond to these changes.

Finally, a third aspect to consider is the mobility of the nodes needing replenishment, and their multi-factored importance. The work in [212] provides an example where mobile nodes must determine which static nodes to replenish. An algorithm to derive which mobile nodes to replenish while considering their importance towards the current objectives of the network would be a valuable contribution. This problem involves reasoning in dynamic situations and in presence of possibly conflicting objectives (i.e., multi-objective optimization). Both of these situations can be successfully handled by means of EAs and SIAs and more generally, through bio-inspired optimizers.

F. Localization

As different types of highly mobile actuators give rise to increasingly dynamic WSANs, localization techniques will have to adapt to this new scenario. These methods do take into account node mobility but they still essentially rely on nodes being fairly slow relative to the localization method's positional estimation speed. In fact, in many of these methods such as [40], [197], [198], it is shown that the reliability of the estimation decreases with anchor nodes moving at higher speeds. Consequently, there exists an untapped research potential for localization methods that take into consideration realistic aspects such as the occurrence of communication network faults and the presence of highly mobile nodes. Dynamic optimization methods fuelled by EAs and SIAs seem a viable alternative in addition to the utilization of FIS and FL to govern the internal control cycle of these mobile anchor nodes.

IX. Conclusions

We have surveyed relevant applications of CI techniques along five different problem categories pertaining to WSANs: *actuation*, *communication*, *sink mobility*, *topology control* and *localization*. The most common type of CI technique employed in WSANs are EAs and SIAs given the overwhelming number of optimization problems (mostly single-objective, a few dealing with multiple conflictive objectives) that can be formulated around these systems and the computational intractability of finding the optimal solution save for small problem instances.

FL and ANNs also show evidence of having been successfully applied, although not as intensively as the aforementioned nature-inspired optimization schemes. As the importance of information uncertainty and WSAN state prediction becomes more paramount, we foresee FL and LSs (ANNs in particular) having a heavier presence in a number of future WSAN papers. We expect to see success stories emerging from the application of other CI methodologies (e.g., *rough set theory* [213]) that are capable of handling other types of uncertainty (e.g. information inconsistency). HSs should also become more visible in this area.

Two of the techniques under the CI umbrella that have been particularly underutilized are RL [55] [56] and Granular Computing [214] [215]. The former provides adaptive learning to changing environmental conditions (a crucial feature in real-world WSANs) whereas the latter is a viable approach to cope with Big Data and Internet of Things requirements by transitioning from a numerically-driven information processing paradigm to a more symbolic, human-centric one. There is a lot of uncharted territory along these lines.

As an important note, it is worth mentioning that the successful execution of these CI methods will depend on the energy resources available in the WSAN. In centralized computation scenarios, this is not a concern as the CI technique will run in a resource-rich node, e.g., a sink or powerful actuator. In localized/distributed computation scenarios, the energy expenses incurred by a node will be a function of several variables, including the complexity of the CI technique itself (see Table I), the frequency with which it is executed or its parametric configuration (e.g., population size for an EA/SA). We are starting to witness a surge in the number of reported applications of CI methods in resource-constrained WSAN environments.

Another emerging research direction is the application of Deep Learning (DL) methods to WSANs, in particular Deep ANNs and Deep RL (hence, DL falls under the LS category in the CI family). Deep Learning is a very vibrant and promising subfield of Machine Learning that has brought groundbreaking success to many application domains. Although some DL applications to WSNs have been recently compiled in [216], we do not see that this disruptive and highly popular technology has invaded the WSAN arena yet.

Finally, we want to point out that the challenges and research opportunities brought about by WSANs share many commonalities with those found in other communication systems, e.g., *Cognitive Radio Sensor Networks* [217] [218]. A worthwhile and promising pursuit would be to investigate the application of CI techniques to these related fields.

References

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, 2004.

[2] A. Nayak and I. Stojmenovic, *Wireless sensor and actuator networks: algorithms and protocols for scalable coordination and data communication*. John Wiley & Sons, 2010.

[3] N. Sabri, S. A. Aljunid, R. Ahmad, M. Malek, A. Yahya, R. Kamaruddin, and M. Salim, "Smart prolong fuzzy wireless sensor-actor network for agricultural application," *Journal of Information Science and Engineering*, vol. 28, no. 2, pp. 295–316, 2012.

[4] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen, "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1013–1024, 2016.

[5] R. Falcon, X. Li, and A. Nayak, "Carrier-based focused coverage formation in wireless sensor and robot networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2406–2417, October 2011.

[6] B. Desjardins, R. Falcon, R. Abielmona, and E. Petriu, "Reliable multiple robot-assisted sensor relocation using multi-objective optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, Canada, July 2016, pp. 4476–4485.

[7] S. Dengler, A. Awad, and F. Dressler, "Sensor/actuator networks in smart homes for supporting elderly and handicapped people," in *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, vol. 2. IEEE, 2007, pp. 863–868.

[8] T. Wark, C. Crossman, W. Hu, Y. Guo, P. Valencia, P. Sikka, P. Corke, C. Lee, J. Henshall, K. Prayaga *et al.*, "The design and evaluation of a mobile sensor/actuator network for autonomous animal control," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 206–215.

[9] R. C. Eberhart and Y. Shi, *Computational Intelligence: concepts to implementations*. Morgan Kaufmann, 2007.

[10] J. Kacprzyk and W. Pedrycz, *Springer handbook of Computational Intelligence*. Springer, 2015.

[11] A. Nayak and I. Stojmenovic, *Wireless sensor and actuator networks: algorithms and protocols for scalable coordination and data communication*. Wiley, 2010.

[12] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, *Wireless sensor and actuator networks: technologies, analysis and design*. Academic Press, 2010.

[13] R. Falcon, "Towards fault reactiveness in wireless sensor networks with mobile carrier robots," Ph.D. dissertation, University of Ottawa, 2012.

[14] N. Mitton and D. Simplot-Ryl, *Wireless sensor and robot networks: from topology control to communication aspects*. World Scientific, 2013.

[15] D.-I. Curiac, "Towards wireless sensor, actuator and robot networks: conceptual framework, challenges and perspectives," *Journal of Network and Computer Applications*, vol. 63, pp. 14–23, 2016.

[16] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[17] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards," *Computer communications*, vol. 30, no. 7, pp. 1655–1695, 2007.

[18] Y.-G. Yue and P. He, "A comprehensive survey on the reliability of mobile wireless sensor networks: Taxonomy, challenges, and future directions," *Information Fusion*, 2018.

[19] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE wireless communications*, vol. 11, no. 6, pp. 6–28, 2004.

[20] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[21] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[22] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, 2009.

[23] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1302–1325, 2011.

[24] I. Khoufi, P. Minet, A. Laouiti, and S. Mahfoudh, "Survey of deployment algorithms in wireless sensor networks: coverage and connectivity issues and challenges," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 10, no. 4, pp. 341–390, 2017.

[25] A. M. Zungeru, L.-M. Ang, and K. P. Seng, "Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison," *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1508–1536, 2012.

[26] F. Dressler and O. B. Akan, "A survey on bio-inspired networking," *Computer Networks*, vol. 54, no. 6, pp. 881–900, 2010.

[27] M. Stojmenovic, "Swarm intelligence for routing in ad hoc wireless networks," *Security and routing in wireless networks*, pp. 167–188, 2005.

[28] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 2, pp. 262–267, 2011.

[29] M. Saleem, G. A. Di Caro, and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions," *Information Sciences*, vol. 181, no. 20, pp. 4597–4624, 2011.

[30] S. Jabbar, R. Iram, A. A. Minhas, I. Shafi, S. Khalid, and M. Ahmad, "Intelligent optimization of wireless sensor networks through bio-inspired computing: survey and future directions," *International Journal of Distributed Sensor Networks*, vol. 9, no. 2, p. 421084, 2013.

[31] M. A. Alsheikh, S. W. Lin, D. Niyato, and H. Tan, "Machine learning in wireless sensor networks: algorithms, strategies, and applications," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1996–2018, 2014.

[32] M. Iqbal, M. Naeem, A. Anpalagan, A. Ahmed, and M. Azam, "Wireless sensor network optimization: multi-objective paradigm," *Sensors*, vol. 15, no. 7, pp. 17 572–17 620, 2015.

[33] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 550–586, 2017.

[34] A. Datta and S. Nandakumar, "A survey on bio inspired meta heuristic based clustering protocols for wireless sensor networks," in *IOP Conference Series: Materials Science and Engineering*, vol. 263, no. 5. IOP Publishing, 2017, p. 052026.

[35] R. V. Kulkarni, A. Forster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 68–96, 2011.

[36] A. Abraham, R. Falcon, and M. Koeppen, *Computational Intelligence in Wireless Sensor Networks: Recent Advances and Future Challenges*. Berlin-Heidelberg, Germany: Springer Verlag, 2017.

[37] X. Peng, Z. Mo, L. Xiao, and G. Liu, "A water-saving irrigation system based on fuzzy control technology and wireless sensor network," *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, 2009.

[38] J. Timmis, L. Murray, and M. Neal, "A neural-endocrine architecture for foraging in swarm robotic systems," *Studies in Computational Intelligence*, vol. 284, pp. 319–330, 2010.

[39] J. S. Liu, S. Y. Wu, and K. M. Chiu, "Path planning of a data mule in wireless sensor network using an improved implementation of clustering-based genetic algorithm," *Proceedings of the IEEE Symposium on Computational Intelligence in Control and Automation*, pp. 30–37, apr 2013.

[40] S. Sivakumar and R. Venkatesan, "Error minimization in localization of wireless sensor networks using modified cuckoo search with mobile anchor positioning (MCS-map) algorithm," *International Journal of Computer Applications*, vol. 95, no. 6, 2014.

[41] M. Hamdy and H. El-Madbouly, "Improvement of QoS management in wireless sensor/actuator networks using fuzzy-genetic approach," *Proceedings of the International Conference on Networking and Media Convergence*, pp. 29–35, mar 2009.

[42] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer, 2003, vol. 53.

[43] K. A. De Jong, "Genetic algorithms are not function optimizers," in *Foundations of genetic algorithms*. Elsevier, 1993, vol. 2, pp. 5–17.

[44] K. Deb and H. Gupta, "Searching for robust Pareto-optimal solutions in multi-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 150–164.

[45] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[46] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[47] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm - a novel tool for complex optimisation problems,"

in *Intelligent Production Machines and Systems*. Elsevier, 2006, pp. 454–459.

[48] S. Das, A. Biswas, S. Dasgupta, and A. Abraham, "Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications," in *Foundations of Computational Intelligence Volume 3*. Springer, 2009, pp. 23–55.

[49] M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 965–997, 2014.

[50] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. IEEE, 2009, pp. 210–214.

[51] L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 103–111, 1996.

[52] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985.

[53] S.-G. Cao, N. W. Rees, and G. Feng, "Mamdani-type fuzzy controllers are universal fuzzy controllers," *Fuzzy Sets and Systems*, vol. 123, no. 3, pp. 359–367, 2001.

[54] I. H. Witten, E. Frank, and M. a. Hall, *Data mining: practical machine learning tools and techniques*, 2011.

[55] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[56] Y. Shoham, R. Powers, and T. Grenager, "Multi-agent reinforcement learning: a critical survey," Technical report, Stanford University, Tech. Rep., 2003.

[57] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, vol. 16, 2016, pp. 2094–2100.

[58] A. Teske, R. Falcon, R. Abielmona, and E. Petriu, "Automating maritime risk assessment with genetic fuzzy systems," in *Proc. of the 2nd International Symposium on Fuzzy and Rough Sets (ISFUROS)*, 2017, pp. 1–10.

[59] S. G. Tzafestas and K. C. Zikidis, "NeuroFAST: On-line neuro-fuzzy ART-based structure and parameter learning TSK model," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 31, no. 5, pp. 797–802, 2001.

[60] H. M. A. Fahmy, *Wireless sensor networks: concepts, applications, experimentation and analysis*. Springer, 2016.

[61] J. McCausland, R. Abielmona, R. Falcon, A. M. Cretu, and E. Petriu, "Auction-based node selection of optimal and concurrent responses for a risk-aware robotic sensor network," *Proceedings of the IEEE International Symposium on Robotic and Sensors Environments*, pp. 136–141, 2013.

[62] J. McCausland, R. Abielmona, R. Falcon, A.-M. Cretu, and E. M. Petriu, "On the role of multi-objective optimization in risk mitigation for critical infrastructures with robotic sensor networks," *Proceedings of the Conference on Genetic and Evolutionary Computation*, pp. 1269–1276, 2014.

[63] R. Falcon, A. Nayak, and I. Stojmenovic, "Robot-assisted wireless sensor networks: recent applications and future challenges," in *Mobile ad hoc networking: the cutting edge directions*, 2nd ed., S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. Wiley, 2014, pp. 737–768.

[64] X. Li, R. Falcon, A. Nayak, and I. Stojmenovic, "Servicing wireless sensor networks by mobile robots," *IEEE Communications Magazine*, vol. 50, pp. 147–154, July 2012.

[65] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[66] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[67] C. Gandhi and V. Arya, "A survey of energy-aware routing protocols and mechanisms for mobile ad hoc networks," in *Intelligent Computing, Networking, and Informatics*. Springer, 2014, pp. 111–117.

[68] R. Falcon, H. Liu, A. Nayak, and I. Stojmenovic, "Controlled straight mobility and energy-aware routing in robotic wireless sensor networks," in *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Hangzhou, China, May 2012, pp. 150–157.

[69] A. Kumar and V. Ranga, "A cluster-based coordination and communication framework using GA for WSANs," *Lecture Notes in Electrical Engineering*, vol. 299 LNEE, pp. 111–124, 2014.

[70] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*. IEEE, 2005, pp. 287a–287a.

[71] M. Ye and Y.-P. Wang, "Coverage repair strategies for wireless sensor networks using mobile actor based on evolutionary computing," *Bulletin of Electrical Engineering and Informatics*, vol. 3, no. 3, pp. 213–222, 2014.

[72] Y. Shi, L. Xie, Y. T. Hou, and H. D. Sherali, "On renewable sensor networks with wireless energy transfer," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1350–1358.

[73] X. Lu, P. Wang, D. Niyato, D. I. Kim, and Z. Han, "Wireless charging technologies: fundamentals, standards, and network applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1413–1452, 2016.

[74] L. Xie, Y. Shi, Y. T. Hou, and H. D. Sherali, "Making sensor networks immortal: an energy-renewal approach with wireless power transfer," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1748–1761, 2012.

[75] Y. Yang and C. Wang, *Wireless rechargeable sensor networks*. Springer, 2015.

[76] L. Militano, M. Erdelj, A. Molinaro, N. Mitton, and A. Iera, "Recharging versus replacing sensor nodes using mobile robots for network maintenance," *Telecommunication Systems*, vol. 63, no. 4, pp. 625–642, 2016.

[77] X. Cui, T. Hardin, R. K. Ragade, and a. S. Elmaghraby, "A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization," *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 194–203, 2004.

[78] F.-k. Chan and C.-y. Wen, "Adaptive AOA / TOA localization using fuzzy particle filter for mobile WSNs," *Scenario*, no. 1, pp. 0–4, 2011.

[79] A. O. De Sá, N. Nedjah, and L. D. M. Mourelle, "Distributed efficient localization in swarm robotics using Min-Max and Particle Swarm Optimization," *Expert Systems with Applications*, vol. 50, pp. 55–65, 2016.

[80] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[81] M. A. Obaidy and A. Ayesh, "Energy efficient algorithm for swarmed sensors networks," *Sustainable Computing: Informatics and Systems*, vol. 5, pp. 54–63, 2015.

[82] J. Lee and G.-L. Park, "Power load distribution for wireless sensor and actuator networks in smart grid buildings," *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1–8, 2013.

[83] H. M. La, T. H. Nguyen, C. H. Nguyen, and H. N. Nguyen, "Optimal flocking control for a mobile sensor network based a moving target tracking," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. IEEE, 2009, pp. 4801–4806.

[84] H. Park, J. Burke, and M. B. Srivastava, "Intelligent lighting control using wireless sensor networks for media production," *Transactions on Internet and Information Systems*, vol. 3, no. 5, pp. 423–443, 2009.

[85] X. Ma, Q. Zhang, and Y. Li, "Genetic algorithm-based multi-robot cooperative exploration," *Proceedings of the IEEE International Conference on Control and Automation*, pp. 1018–1023, 2007.

[86] B. Desjardins, R. Falcon, R. Abielmona, and E. Petriu, "A multi-objective optimization approach to reliable robot-assisted sensor relocation," *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 956–964, may 2015.

[87] ——, "Planning robust sensor relocation trajectories for a mobile robot with evolutionary multi-objective optimization," in *Computational Intelligence in Wireless Sensor Networks: Recent Advances and Future Challenges*, A. Abraham, R. Falcon, and M. Koeppen, Eds. Springer Verlag, 2017, ch. 8, pp. 179–210.

[88] ——, "Reliable multiple robot-assisted sensor relocation using multi-objective optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, Canada, July 2016, pp. 4476–4485.

[89] N. Primeau, R. Abielmona, R. Falcon, and E. Petriu, "Maritime smuggling detection and mitigation using risk-aware hybrid robotic sensor networks," in *Cognitive and Computational Aspects of Situation Management (CogSIMA), 2017 IEEE Conference on*. IEEE, 2017, pp. 1–7.

[90] G.-p. Qi, P. Song, and K.-j. Li, "Blackboard mechanism based ant colony theory for dynamic deployment of mobile sensor networks," *Journal of Bionic Engineering*, vol. 5, no. 3, pp. 197–203, 2008.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2018.2850220, IEEE Communications Surveys & Tutorials

[91] F. b. Li, J. Fu, D. Du, and W. Bao, "Adaptive distributed event-triggered collaborative control with WSAN," *Communications in Computer and Information Science*, vol. 462, pp. 523–532, 2014.

[92] D. Liu, X. Zhou, A. Liang, and H. Guan, "A swarm intelligence based algorithm for distribute search and collective cleanup," in *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*, vol. 2. IEEE, 2010, pp. 161–165.

[93] X. Li and H.-x. Ma, "Particle swarm optimization based multi-robot task allocation using wireless sensor network," in *Information and Automation, 2008. ICIA 2008. International Conference on*. IEEE, 2008, pp. 1300–1303.

[94] J. Yang, H. Zhang, Y. Ling, C. Pan, and W. Sun, "Task allocation for wireless sensor network using modified binary particle swarm optimization," *IEEE Sensors Journal*, vol. 14, no. 3, pp. 882–892, 2014.

[95] G. K. Venayagamoorthy and S. Doctor, "Navigation of mobile sensors using PSO and embedded PSO in a fuzzy logic controller," in *Industry Applications Conference, 2004*, vol. 2. IEEE, 2004, pp. 1200–1206.

[96] H. Xue, H. Qin, X. Li, and H. Ma, "Swarm intelligence based WSN-mediated distributed multi-robot task allocation," *Proceedings of the Chinese Control Conference*, pp. 451–456, 2008.

[97] S. B. Kumar and B. A. Prakash, "Artificial bee colony algorithm for multi-target tracking in mobility sensor networks," in *Electronics and Communication Systems (ICECS), 2014 International Conference on*. IEEE, 2014, pp. 1–5.

[98] F. Mourad, H. Chehade, H. Snoussi, F. Yalaoui, L. Amodeo, and C. Richard, "Controlled mobility sensor networks for target tracking using ant colony optimization," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1261–1273, aug 2012.

[99] M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, "A PSO multi-robot exploration approach over unreliable MANETs," *Advanced Robotics*, vol. 27, no. December, pp. 1221–1234, 2013.

[100] N. Sabri, S. A. Aljunid, R. B. Ahmad, M. F. Malek, A. Yahya, R. Kamaruddin, and M. S. Salim, "Smart prolong fuzzy wireless sensor-actor network for agricultural application," *Journal of Information Science and Engineering*, vol. 28, no. 2, pp. 295–316, 2012.

[101] N. Sabri, S. A. Aljunid, M. S. Salim, R. Kamaruddin, R. Badlishah Ahmad, and M. F. Malek, "Cognitive wireless sensor actor network: an agricultural perspective," *International Journal of Innovative Computing, Information and Control*, vol. 10, no. 2, pp. 631–658, 2014.

[102] Y. Wang and Z. Wang, "Design of intelligent residential lighting control system based on zigbee wireless sensor network and fuzzy controller," in *Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference on*. IEEE, 2010, pp. 561–564.

[103] P. Song, J. Li, K. Li, and J. Liang, "Hybrid multi-robot path planning strategy based on RRT and fuzzy control in wireless sensor network," in *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*. IEEE, 2009, pp. 877–881.

[104] D. Elmazi, T. Inaba, S. Sakamoto, T. Oda, M. Ikeda, and L. Barolli, "Selection of secure actors in wireless sensor and actor networks using fuzzy logic," in *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2015 10th International Conference on*. IEEE, 2015, pp. 125–131.

[105] D. Elmazi, E. Kulla, T. Oda, E. Spaho, S. Sakamoto, and L. Barolli, "A comparison study of two fuzzy-based systems for selection of actor node in wireless sensor actor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 5, pp. 635–645, 2015.

[106] D. Elmazi, S. Sakamoto, T. Oda, E. Kulla, E. Spaho, and L. Barolli, "Two fuzzy-based systems for selection of actor nodes in wireless sensor and actor networks: a comparison study considering security parameter effect," *Mobile Network Applications*, vol. 21, no. 1, pp. 53–64, Feb. 2016.

[107] K. Ebisu, T. Inaba, D. Elmazi, M. Ikeda, L. Barolli, and E. Kulla, "A fuzzy-based testbed design for wireless sensor and actuator networks," in *Network-Based Information Systems (NBiS), 2015 18th International Conference on*. IEEE, 2015, pp. 548–553.

[108] D. Elmazi, M. Cuka, K. Bylykbashi, E. Spaho, M. Ikeda, and L. Barolli, "Selection of actor nodes in wireless sensor and actor networks considering actor-sensor coordination quality parameter," in *International Conference on Broadband and Wireless Computing, Communication and Applications*. Springer, 2017, pp. 87–99.

[109] ——, "Selection of actor nodes in wireless sensor and actor networks considering failure of assigned task as new parameter," in *International Conference on Emerging Internetworking, Data & Web Technologies*. Springer, 2018, pp. 106–118.

[110] D. Elmazi, M. Cuka, T. Oda, M. Ikeda, and L. Barolli, "Effect of node density on actor selection in WSANs: a comparison study for two fuzzy-based systems," in *Advanced Information Networking and Applications (AINA), 2017 IEEE 31st International Conference on*. IEEE, 2017, pp. 865–871.

[111] D. Elmazi, M. Cuka, E. Kulla, T. Oda, M. Ikeda, and L. Barolli, "Implementation and comparison of two intelligent systems based on fuzzy logic for actor selection in WSANs: effect of node density on actor selection," *International Journal of Space-Based and Situated Computing*, vol. 7, no. 4, pp. 229–238, 2017.

[112] G. K. Venayagamoorthy, L. L. Grant, and S. Doctor, "Collective robotic search using hybrid techniques: fuzzy logic and swarm intelligence inspired by nature," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 3, pp. 431–441, 2009.

[113] D. Pham, M. Awadalla, and E. Eldukhri, "Fuzzy and neuro-fuzzy based co-operative mobile robots," in *Intelligent Production Machines and Systems*. Elsevier, 2006, pp. 578–583.

[114] A. Jankowska, M. C. Schut, and N. Ferreira-Schut, "A wireless actuator-sensor neural network for evacuation routing," *Proceedings of the International Conference on Sensor Technologies and Applications*, pp. 139–144, 2009.

[115] J. I. Z. Chen, C. C. Yu, M. T. Hsieh, and Y. N. Chung, "Employing CHNN to develop a data refining algorithm for wireless sensor networks," *Proceedings of the World Congress on Computer Science and Information Engineering*, vol. 1, pp. 24–31, 2009.

[116] J. Heinerman, A. Zonta, E. Haasdijk, and A. E. Eiben, "On-line evolution of foraging behaviour in a population of real robots," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2016, pp. 198–212.

[117] M. Duarte, V. Costa, J. Gomes, T. Rodrigues, F. Silva, S. M. Oliveira, and A. L. Christensen, "Evolution of collective behaviors for a real swarm of aquatic surface robots," *PloS one*, vol. 11, no. 3, p. e0151834, 2016.

[118] K. Ota, M. Dong, Z. Cheng, J. Wang, X. Li, and X. Shen, "ORACLE: Mobility control in wireless sensor and actor networks," *Computer Communications*, vol. 35, no. 9, pp. 1029–1037, 2012.

[119] M. Dong, K. Ota, H. Li, S. Du, H. Zhu, and S. Guo, "RENDEZVOUS: towards fast event detecting in wireless sensor and actor networks," *Computing*, vol. 96, no. 10, pp. 995–1010, 2014.

[120] J. McCausland, R. Abielmona, R. Falcon, and E. Petriu, "Risk-aware sensor networks for critical infrastructure monitoring," in *11th International Conference on Shock & Impact Loads on Structures (SILOS), Ottawa, Canada*, 2015, pp. 295–304.

[121] Y. Li and L. E. Parker, "Intruder detection using a wireless sensor network with an intelligent mobile robot response," in *SouthEastCon, 2008. IEEE*. IEEE, 2008, pp. 37–42.

[122] S. Kernbach, H. Hamann, J. Stradner, R. Thenius, T. Schmickl, K. Crailsheim, A. C. Van Rossum, M. Sebag, N. Bredeche, Y. Yao, G. Baele, Y. Van De Peer, J. Timmis, M. Mohktar, A. Tyrrell, A. E. Eiben, S. P. McKibbin, W. Liu, and A. F. T. Winfield, "On adaptive self-organization in artificial robot organisms," *Proceedings of Computation World: Future Computing, Service Computation, Adaptive, Content, Cognitive, Patterns*, pp. 33–43, 2009.

[123] X. Han and M. Hong-xu, "Bio-inspired stochastic chance-constrained multi-robot task allocation using WSN," in *Proc. of the 2008 IEEE International Joint Conference on Neural Networks*. IEEE, 2008, pp. 721–726.

[124] R. Falcon, R. Abielmona, and A. Nayak, "An evolving risk management framework for wireless sensor networks," in *Proceedings of the 2011 IEEE Int'l Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA)*, Ottawa, Canada, September 2011, pp. 1–6.

[125] R. Falcon and R. Abielmona, "A response-aware risk management framework for search-and-rescue operations," in *2012 IEEE Congress on Evolutionary Computation (CEC)*, Brisbane, Australia, June 2012, pp. 1540–1547.

[126] S. Sarangi and S. Kar, "Genetic algorithm based mobility aware clustering for energy efficient routing in wireless sensor networks," *Proceedings of the IEEE International Conference on Networks*, pp. 1–6, dec 2011.

[127] H. Yang, F. Ye, and B. Sikdar, "A swarm-intelligence-based protocol for data acquisition in networks with mobile sinks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 8, pp. 931–945, 2008.

[128] Y.-F. Hu, Y.-S. Ding, L.-H. Ren, K.-R. Hao, and H. Han, "An endocrine cooperative particle swarm optimization algorithm for routing recovery problem of wireless sensor networks with multiple mobile sinks," *Information Sciences*, vol. 300, pp. 100–113, 2015.

[129] J. Yang, F. Liu, J. Cao, and L. Wang, "Discrete particle swarm optimization routing protocol for wireless sensor networks with multiple mobile sinks," *Sensors*, vol. 16, no. 7, p. 1081, 2016.

[130] S. C. Oliveira, G. A. Duarte, and H. B. Cunha, "Unified coordination-communication strategy to swarm controlled mobile wireless sensor network," *IEEE Latin America Transactions*, vol. 12, no. 5, pp. 951–956, aug 2014.

[131] K. Nawaz, I. Petrov, and A. P. Buchmann, "Configurable, energy-efficient, application- and channel-aware middleware approaches for cyber-physical systems," *Studies in Computational Intelligence*, vol. 540, pp. 3–65, 2014.

[132] F. Xia, W. Zhao, Y. Sun, and Y.-C. Tian, "Fuzzy logic control based QoS management in wireless sensor/actuator networks," *Sensors*, vol. 7, no. 12, pp. 3179–3191, 2007.

[133] T. Oda, D. Elmazi, M. Cuka, E. Kulla, M. Ikeda, and L. Barolli, "Performance evaluation of a deep q-network based simulation system for actor node mobility control in wireless sensor and actor networks considering three-dimensional environment," in *International Conference on Intelligent Networking and Collaborative Systems*. Springer, 2017, pp. 41–52.

[134] X. Wu, Y. Wang, and Y. Hu, "A swarm intelligent algorithm based route maintaining protocol for mobile sink wireless sensor networks," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[135] J. R. Srivastava and T. Sudarshan, "A genetic fuzzy system based optimized zone based energy efficient routing protocol for mobile sensor networks (OZEEP)," *Applied Soft Computing*, vol. 37, pp. 863–886, 2015.

[136] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, "Mobile sink-based adaptive immune energy-efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4576–4586, 2015.

[137] W. Cai, M. Chen, T. Hara, and L. Shu, "GA-MIP: genetic algorithm based multiple mobile agents itinerary planning in wireless sensor networks," in *Wireless Internet Conference (WICON), 2010 The 5th Annual ICST*. IEEE, 2010, pp. 1–8.

[138] M. Alnuaimi, K. Shuaib, K. Alnuaimi, and M. Abdel-Hafez, "Data gathering in delay tolerant wireless sensor networks using a ferry," *Sensors*, vol. 15, no. 10, p. 25809, 2015.

[139] J.-h. Zhong and J. Zhang, "Ant colony optimization algorithm for lifetime maximization in wireless sensor network with mobile sink," *Proceedings of the Conference on Genetic and Evolutionary Computation*, pp. 1199–1204, 2012.

[140] D.-T. Ho, E. I. Grøtli, P. B. Sujit, T. A. Johansen, and J. B. Sousa, "Optimization of wireless sensor network and UAV data acquisition," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 1, pp. 159–179, 2015.

[141] N. A. A. Latiff, N. M. A. Latiff, and R. B. Ahmad, "Prolonging lifetime of wireless sensor networks with mobile base station using particle swarm optimization," in *Modeling, Simulation and Applied Optimization (ICMSAO), 2011 4th International Conference on*. IEEE, 2011, pp. 1–6.

[142] E. M. Saad, M. H. Awadalla, and R. R. Darwish, "A data gathering algorithm for a mobile sink in large-scale sensor networks," *Proceedings of the International Conference on Wireless and Mobile Communications*, pp. 207–213, 2008.

[143] Y. Yue, J. Li, H. Fan, and Q. Qin, "Optimization-based artificial bee colony algorithm for data collection in large-scale mobile wireless sensor networks," *Journal of Sensors*, vol. 2016, 2016.

[144] G. Comarela, K. Gonçalves, G. L. Pappa, J. Almeida, and V. Almeida, "Robot routing in sparse wireless sensor networks with continuous ant colony optimization," *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 599–605, 2011.

[145] G. Kavitha, "Bacterial Foraging Optimization (BFO) based Traveling Salesman Problem (TSP) for data collection in mobile sinks," *International Journal of Scientific and Computational Intelligence*, vol. 2, no. 1, 2015.

[146] H. M. Sarvestani, M. A. Sarram, V. Derhami, and M. Rezaei, "Improving network lifetime with fuzzy sink mobility scheme in wireless sensor network."

[147] J. Faigl and G. A. Hollinger, "Autonomous data collection using a self-organizing map," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1703–1715, 2018.

[148] M. A. Uddin, A. Mansour, D. L. Jeune, M. Ayaz, and e.-H. M. Aggoune, "UAV-assisted dynamic clustering of wireless sensor networks for crop health monitoring," *Sensors*, vol. 18, no. 2, p. 555, 2018.

[149] S. Shi, X. Liu, and X. Gu, "An energy-efficiency Optimized LEACH-C for wireless sensor networks," in *Communications and Networking in China (CHINACOM), 2012 7th International ICST Conference on*. IEEE, 2012, pp. 487–492.

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

[151] Y. l. Zeng and J. h. Zheng, "A genetic algorithm-based wireless sensor network coverage-enhancing approach," *Proceedings of the International Conference on Natural Computation*, pp. 1069–1073, may 2012.

[152] B. A. Attea, E. A. Khalil, and A. Cosar, "Multi-objective evolutionary routing protocol for efficient coverage in mobile sensor networks," *Soft Computing*, vol. 19, no. 10, pp. 2983–2995, 2015.

[153] R. Kuawattanaphan, T. Kumrai, and P. Champrasert, "Wireless sensor nodes redeployment using a multiobjective optimization evolutionary algorithm," *Proceedings of the IEEE International Conference of IEEE Region 10*, pp. 1–6, 2013.

[154] A. A. Bara'a, E. A. Khalil, and S. Özdemir, "Biologically inspired probabilistic coverage for mobile sensor networks," *Soft Computing*, vol. 18, no. 11, pp. 2313–2322, 2014.

[155] X. Yu, C. Wu, D. Chen, and N. Hu, "Level set based coverage holes detection and holes healing scheme in hybrid sensor network," *International Journal of Distributed Sensor Networks*, vol. 9, no. 11, pp. 1–9, 2013.

[156] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, "Rearrangement of mobile wireless sensor nodes for coverage maximization based on immune node deployment algorithm," *Computers & Electrical Engineering*, vol. 43, pp. 76–89, 2015.

[157] ——, "Coverage maximization in mobile wireless sensor networks utilizing immune node deployment algorithm," *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, 2014.

[158] L. Kuang and Z. Cai, "Immune system based redeployment scheme for wireless sensor networks," *Proceedings of IET International Conference on Wireless Sensor Network*, pp. 69–72, 2010.

[159] R. Katsuma, Y. Murata, N. Shibata, K. Yasumoto, and M. Ito, "Extending k-coverage lifetime of wireless sensor networks using mobile sensor nodes," in *Wireless and Mobile Computing, Networking and Communications, 2009. WIMOB 2009. IEEE International Conference on*. IEEE, 2009, pp. 48–54.

[160] N. Rahmani and F. Nematy, "EAVD: An evolutionary approach based on Voronoi diagram for node deployment in wireless sensor networks," *Advances in Intelligent and Soft Computing*, vol. 130 AISC, no. VOL. 1, pp. 121–129, 2012.

[161] O. Banimelhem, M. Mowafi, and W. Aljoby, "Genetic algorithm based node deployment in hybrid wireless sensor networks," *Communications and Network*, vol. 2013, 2013.

[162] X. Jiang, Y. P. Chen, and T. Yu, "Localized distributed sensor deployment via coevolutionary computation," *Proceedings of the International Conference on Communications and Networking in China*, pp. 784–788, 2008.

[163] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "Dynamic deployment of wireless sensor networks by biogeography based optimization algorithm," *Journal of Sensor and Actuator Networks*, vol. 1, no. 2, pp. 86–96, 2012.

[164] S. Singh and A. Kumar, "Novel optimal deployment of sensor nodes using bio inspired algorithm," *Proceedings of the International Conference on Advanced Communication Control and Computing Technologies*, pp. 847–851, may 2014.

[165] R. Shams and F. H. Khan, "Solving wireless network scheduling problem by genetic algorithm," *Global Engineers and Technologist Review*, vol. 2, no. 11, 2012.

[166] I. Saadallah, "Autonomous or assisted deployment by mobile robots of wireless sensor networks: coverage and connectivity issues," Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2015.

[167] A. Peiravi, H. R. Mashhadi, and S. Hamed Javadi, "An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm," *International Journal of Communication Systems*, vol. 26, no. 1, pp. 114–126, 2013.

[168] M. Azharuddin and P. K. Jana, "A GA-based approach for fault tolerant relay node placement in wireless sensor networks," in *Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on*. IEEE, 2015, pp. 1–6.

[169] A. J. Perez, M. A. Labrador, and P. M. Wightman, "A multiobjective approach to the relay placement problem in wsns," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*. IEEE, 2011, pp. 475–480.

[170] W. Yu, X. Li, H. Yang, and B. Huang, "A multi-objective metaheuristics study on solving constrained relay node deployment problem in WSNs," *Intelligent Automation & Soft Computing*, pp. 1–10, 2017.

[171] J. M. Lanza-Gutierrez and J. A. Gomez-Pulido, "Studying the multiobjective variable neighbourhood search algorithm when solving the relay node placement problem in wireless sensor networks," *Soft Computing*, vol. 20, no. 1, pp. 67–86, 2016.

[172] W. Yiyue, L. Hongmei, and H. Hengyang, "Wireless sensor network deployment using an optimized artificial fish swarm algorithm," in *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, vol. 2. IEEE, 2012, pp. 90–94.

[173] F. Nematy, A. M. Rahmani, M. Teshnelab, and N. Rahmani, "Ant colony based node deployment and search in wireless sensor networks," in *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on*. IEEE, 2010, pp. 363–366.

[174] A. Ray and D. De, "An energy efficient sensor movement approach using multi-parameter reverse glowworm swarm optimization algorithm in mobile wireless sensor network," *Simulation Modelling Practice and Theory*, vol. 62, pp. 117–136, 2016.

[175] Q. Ni, H. Du, Q. Pan, C. Cao, and Y. Zhai, "An improved dynamic deployment method for wireless sensor network based on multi-swarm particle swarm optimization," *Natural Computing*, pp. 1–9, 2015.

[176] N. A. B. A. Aziz, A. W. Mohemmed, and B. D. Sagar, "Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization," *Proceedings of the International Conference on Intelligent and Advanced Systems*, pp. 961–965, 2007.

[177] Z. Li and L. Lei, "Sensor node deployment in wireless sensor networks based on improved particle swarm optimization," *Proceedings of the International Conference on Applied Superconductivity and Electromagnetic Devices*, pp. 215–217, 2009.

[178] X. Wang, S. Wang, and J.-J. Ma, "An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment," *Sensors*, vol. 7, no. 3, pp. 354–370, 2007.

[179] S. M. A. Salehizadeh, A. Dirafzoon, M. B. Menhaj, and A. Afshar, "Coverage in wireless sensor networks based on individual particle optimization," *Proceedings of the International Conference on Networking, Sensing and Control*, pp. 501–506, apr 2010.

[180] Z. Xiao, J. Yan, X. Yan, G. Cheng, and X. Qiang, "A new optimal deployment algorithm for actor node in WSAN," in *Network and Information Systems for Computers (ICNISC), 2015 International Conference on*. IEEE, 2015, pp. 532–535.

[181] C. Ozturk, D. Karaboga, and B. Gorkemli, "Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm." *Sensors*, vol. 11, no. 6, pp. 6056–65, 2011.

[182] R. V. Kulkarni and G. K. Venayagamoorthy, "Bio-inspired algorithms for autonomous deployment and localization of sensor nodes," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 40, no. 6, pp. 663–675, 2010.

[183] R. Falcon, X. Li, A. Nayak, and I. Stojmenovic, "The one-commodity traveling salesman problem with selective pickup and delivery: an ant colony approach," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Barcelona, Spain, 2010, pp. 4326–4333.

[184] L.-M. Mou and X.-L. Dai, "A novel ant colony system for solving the one-commodity traveling salesman problem with selective pickup and delivery," in *Natural Computation (ICNC), 2012 Eighth International Conference on*. IEEE, 2012, pp. 1096–1101.

[185] Y. Wang, A. Barnawi, R. F. de Mello", and I. Stojmenovic, "Localized ant colony of robots for redeployment in wireless sensor networks," *Journal of Multi-Valued Logic and Soft Computing*, vol. 23, pp. 35–51, 2014.

[186] L. Cheng, Y. Wang, C. Wu, and Q. Han, "A PSO-based maintenance strategy in wireless sensor networks," *Intelligent Automation & Soft Computing*, vol. 21, no. 1, pp. 65–75, 2015.

[187] B. J. Farahani, H. Ghaffarian, and M. Fathy, "A fuzzy based priority approach in mobile sensor network coverage," *International Journal of Recent Trends in Engineering*, vol. 2, no. 1, 2009.

[188] J. Li, K. Li, and W. Zhu, "Improving sensing coverage of wireless sensor networks by employing mobile robots," *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 899–903, dec 2007.

[189] R. Falcon, X. Li, A. Nayak, and I. Stojmenovic, "A harmony-seeking firefly swarm to the periodic replacement of damaged sensors by a team of mobile robots," in *2012 IEEE International Conference on Communications (ICC)*, Ottawa, Canada, June 2012, pp. 6436–6440.

[190] S. M. A. Salehizadeh, P. Yadmellat, and M. B. Menhaj, "Transition from particle swarm optimization to individual particle optimization," in *Control Applications,(CCA) & Intelligent Control,(ISIC), 2009 IEEE*. IEEE, 2009, pp. 1409–1414.

[191] S. Phatak and S. S. Rao, "Logistic map: a possible random-number generator," *Physical review E*, vol. 51, no. 4, p. 3670, 1995.

[192] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[193] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[194] W. Khatib and P. J. Fleming, "The stud GA: a mini revolution?" *Parallel Problem Solving from Nature*, pp. 683–691, 1998.

[195] J. Huanxiang, W. Yong, and T. Xiaoling, "Localization algorithm for mobile anchor node based on genetic algorithm in wireless sensor network," *Proceedings of the International Conference Intelligent Computing and Integrated Systems*, pp. 40–44, 2010.

[196] S. R. Karedla and S. Anuradha, "Localization error minimization using GA in MWSN," *International Journal of Advanced Electrical and Electronics Engineering*, vol. 2, no. 4, pp. 17–24, 2013.

[197] S. Sivakumar, R. Venkatesan, and M. Karthiga, "Evolutionary approaches for minimising error in localisation of wireless sensor networks," *International Journal of Sensor Networks*, vol. 17, no. 1, pp. 17–26, 2015.

[198] S. Sivakumar and R. Venkatesan, "Performance evaluation of hybrid evolutionary algorithms in minimizing localization error for wireless sensor networks," *Journal of Scientific & Industrial Research*, vol. 75, pp. 289–295, 2016.

[199] H. Bao, B. Zhang, C. Li, and Z. Yao, "Mobile anchor assisted particle swarm optimization (PSO) based localization algorithms for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 12, no. 15, pp. 1313–1325, 2012.

[200] F. Qin, C. Wei, and L. Kezhong, "Node localization with a mobile beacon based on ant colony algorithm in wireless sensor networks," in *Communications and Mobile Computing (CMC), 2010 International Conference on*, vol. 3. IEEE, 2010, pp. 303–307.

[201] D. Herrero and H. Martínez, "Range-only fuzzy Voronoi-enhanced localization of mobile robots in wireless sensor networks," *Robotica*, vol. 30, no. 07, pp. 1063–1077, 2011.

[202] R. Dutta, S. Saha, and A. K. Mukhopadhyay, "Tracking heterogeneous dynamic sensor node using fuzzy logic to prolong system lifetime in WSN," *Procedia Engineering*, vol. 38, pp. 522–527, 2012.

[203] M. Gholami, N. Cai, and R. Brennan, "An artificial neural network approach to the problem of wireless sensors network localization," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp. 96–109, 2013.

[204] N. Irfan, M. Bolic, M. C. E. Yagoub, and V. Narasimhan, "Neural-based approach for localization of sensors in indoor environment," *Telecommunication Systems*, vol. 44, no. 1-2, pp. 149–158, 2010.

[205] Q. Guo and T. Tang, "Node cooperation based support vector machine localization algorithm in mobile wireless sensor networks," *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*, pp. 79 820N—-79 820N, 2011.

[206] R. Qi, S. Li, T. Ma, and F. Qian, "Localization with a mobile anchor using ABC-GA hybrid algorithm in wireless sensor networks," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 16, no. 6, pp. 741–747, 2012.

[207] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, 2013.

[208] A. Egorova-Foerster and A. L. Murphy, "A feedback-enhanced learning approach for routing in WSN," *Proceedings of the ITG-GI Conference on Communication in Distributed Systems*, pp. 1–12, Feb 2007.

[209] L. Song, D. Niyato, Z. Han, and E. Hossain, "Game-theoretic resource allocation methods for device-to-device communication," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 136–144, 2014.

[210] M. Egorov, "Multi-Agent Deep Reinforcement Learning," *CS231n*, p. 8, 2016.

[211] N. Pavlidis, K. E. Parsopoulos, and M. N. Vrahatis, "Computing nash equilibria through computational intelligence methods," *Journal of Computational and Applied Mathematics*, vol. 175, no. 1, pp. 113–136, 2005.

[212] M. Rahimi, H. Shah, G. S. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *Proc. of the IEEE International Conference on Robotics and Automation, 2003*, vol. 1. IEEE, 2003, pp. 19–24.

[213] A. Abraham, R. Falcon, and R. Bello, *Rough set theory: a true landmark in data analysis*. Berlin-Heidelberg, Germany: Springer Verlag, 2009.

[214] W. Pedrycz, A. Skowron, and V. Kreinovich, *Handbook of granular computing*. John Wiley & Sons, 2008.

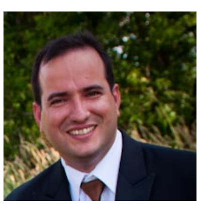
[215] R. Bello, R. Falcon, W. Pedrycz, and J. Kacprzyk, *Granular computing: at the junction of rough sets and fuzzy sets*. Berlin-Heidelberg, Germany: Springer Verlag, 2008.

[216] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[217] A. Ahmad, S. Ahmad, M. H. Rehmani, and N. U. Hassan, "A survey on radio resource allocation in cognitive radio sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 888–917, 2015.

[218] S. H. R. Bukhari, M. H. Rehmani, and S. Siraj, "A survey of channel bonding for wireless networks and guidelines of channel bonding for futuristic cognitive radio sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 924–948, 2016.

Nicolas Primeau received his B.A.Sc. (2015) and M.A.Sc. (2017) in Computer and Electrical Engineering from the School of Electrical Engineering and Computer Science at the University of Ottawa. His research interests included wireless sensor networks, computational intelligence, decision support systems for risk management, and multiobjective optimization. He joined Amazon, Inc. in 2017 where he is currently involved in developing Amazon's Alexa services.



Rafael Falcon (M'09-SM'14) received his PhD degree in Computer Science from the School of Information Technology and Engineering at the University of Ottawa in 2012. Previously, he attained his Bachelor and MSc degrees in Computer Science from the Central University of Las Villas, Cuba. He currently works as a Research Scientist for Larus Technologies Corporation, an Ottawa-based firm that specializes in predictive analytics, high-level information fusion and decision support from a Computational Intelligence angle. He is also an Adjunct Professor with the School of Electrical Engineering and Computer Science, University of Ottawa. His research interests are in the area of Computational Intelligence with applications to security and defense, including wireless sensor networks, robotics, maritime domain awareness, predictive analytics, machine learning and multi-sensor data fusion.

Rafael is an IEEE senior member and a member of the International Rough Sets Society and the Canadian Artificial Intelligence Association. He also serves as the vice-chair of the IEEE Computational Intelligence Society's Security, Surveillance and Defense Task Force and the chair of the Ottawa IEEE Computational Intelligence/ Systems, Man and Cybernetics joint chapter.

Rafael is a recipient of the 2011 IEEE CIS "*Walter Karplus*" Graduate Student Research Grant, the National Sciences & Engineering Research Council of Canada's Industrial Research & Development Fellowship, the Ontario Centres of Excellence's First Job Award, the 2009-2010 Ontario Graduate Scholarship, the 2014 MITACS and NRC-IRAP Award for Commercialization and two MITACS Accelerate PhD Fellowships.

Rami Abielmona (M'99-SM'11) received his B.A.Sc., M.A.Sc. and Ph.D. degrees in Computer and Electrical Engineering, from the School of Information Technology and Engineering at the University of Ottawa, where he currently serves as an Adjunct Professor. He is the Vice President of Research & Engineering at Larus Technologies Corporation where he is responsible for all research and development of software and embedded products, as well as the management and direction of the research team. His research interests include computational intelligence, sensor networking, high-level information fusion and computationally intelligent processing architectures. Rami is also a co-founder of the Canadian Tracking and Fusion Group (CTFG) and currently serves as the Chair of the IEEE Computational Intelligence Society Security, Surveillance and Defense Applications Task Force.

Rami was a CMC Success Story in 2005 for his Ph.D. research. He received the NSERC Industrial Research and Development Fellowship (IR&DF) in 2007 and the IEEE MGA Achievement Award in 2008. He was one of the recipients of the Ottawa Business Journal (OBJ) Top 40 Under 40 Award in 2011. He was also named as the Part-Time Professor of the Year at both the Faculty of Engineering and the University of Ottawa in 2012. He was awarded the IEEE Ottawa Section Outstanding Volunteer Award in 2014. He was also a recipient of the NSERC Synergy Award for Innovation (for Small and Medium-Sized Companies) in 2016. He also became a licensed Professional Engineer in Ontario in 2008.

Emil M. Petriu (M'86-SM'88-F'01-LF'17) is a Professor and University Research Chair in the School of Electrical Engineering and Computer Science at the University of Ottawa, Canada. His research interests include biology-inspired robot sensing and perception, soft computing, human-computer interaction, and system-on-chip design. During his career he has published over 530 papers in peer-reviewed international journals and conference proceedings, 27 book chapters, co-authored two books, edited other four books, and received two patents. He is a Life Fellow of IEEE, Fellow of the Canadian Academy of Engineering, and Fellow of the Engineering Institute of Canada. He is a co-recipient of the 2003 IEEE's Donald G. Fink Prize Paper Award and a received the 2003 IEEE Instrumentation and Measurement Society Technical Award and the 2009 IEEE Instrumentation and Measurement Society Distinguished Service Award.