Decision Making in Multiagent Systems: A Survey

Yara Rizk, *Member, IEEE*, Mariette Awad, *Member, IEEE*, and Edward W. Tunstel, *Fellow, IEEE*

***Abstract*—Intelligent transport systems, efficient electric grids, and sensor networks for data collection and analysis are some examples of the multiagent systems (MAS) that cooperate to achieve common goals. Decision making is an integral part of intelligent agents and MAS that will allow such systems to accomplish increasingly complex tasks. In this survey, we investigate state-of-the-art work within the past five years on cooperative MAS decision making models, including Markov decision processes, game theory, swarm intelligence, and graph theoretic models. We survey algorithms that result in optimal and suboptimal policies such as reinforcement learning, dynamic programming, evolutionary computing, and neural networks. We also discuss the application of these models to robotics, wireless sensor networks, cognitive radio networks, intelligent transport systems, and smart electric grids. In addition, we define key terms in the area and discuss remaining challenges that include incorporating big data advancements to decision making, developing autonomous, scalable and computationally efficient algorithms, tackling more complex tasks, and developing standardized evaluation metrics. While recent surveys have been published on this topic, we present a broader discussion of related models and applications.**

***Note to Practitioners:* Future smart cities will rely on cooperative MAS that make decisions about what actions to perform that will lead to the completion of their tasks. Decision making models and algorithms have been developed and reported in the literature to generate such sequences of actions. These models are based on a wide variety of principles including human decision making and social animal behavior. In this paper, we survey existing decision making models and algorithms that generate optimal and suboptimal sequences of actions. We also discuss some of the remaining challenges faced by the research community before more effective MAS deployment can be achieved in this age of Internet of Things, robotics, and mobile devices. These challenges include developing more scalable and efficient algorithms, utilizing the abundant sensory data available, tackling more complex tasks, and developing evaluation standards for decision making.**

***Index Terms*—Cooperation, decision making models, game theory, Markov decision process (MDP), multiagent systems (MASs), swarm intelligence.**

Manuscript received November 25, 2017; revised March 12, 2018; accepted May 17, 2018. Date of publication May 25, 2018; date of current version September 7, 2018. This work was supported in part by the National Council of Scientific Research in Lebanon, and in part by the University Research Board at the American University of Beirut. *(Corresponding author: Mariette Awad.)*

Y. Rizk and M. Awad are with the Department of Electrical and Computer Engineering, American University of Beirut, Beirut 1107 2020, Lebanon (e-mail: yar01@aub.edu.lb; mariette.awad@aub.edu.lb).

E. W. Tunstel was with the Intelligent Systems Center, Johns Hopkins Applied Physics Laboratory, Laurel, MD 20723 USA. He is now with the Systems Department, United Technologies Research Center, East Hartford, CT 06108 USA (e-mail: tunstel@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCDS.2018.2840971

I. INTRODUCTION

THE NUMBER of devices connected to the Internet has been increasing over the past few years and projected to exceed 20 billion devices by 2020 [1], [2]. These devices can communicate with each other to form multiagent systems (MASs) that can cooperate to overcome individual limitations and achieve complex tasks. This has led to the emergence of cognitive computing systems which were defined by IBM as systems that can interact with each other and humans to exploit their strengths when accomplishing a task [3]. At the heart of cognitive systems is the decision making, or planning and control module which allows agents to generate a sequence of actions that will lead to the accomplishment of their goals.

Multiple surveys on MAS decision making have been published. While some briefly discussed a wide range of models and applications, others focused on a specific model or application. We present a more up-to-date discussion on cooperative MAS decision making, covering a wide range of applications and models. Fig. 1 depicts the scope of existing surveys in terms of their relative breadth of covered models and applications, while highlighting the targeted scope of our survey. Color coding distinguishes references' publication date: surveys published more than ten years ago are in red, 5–10 years in green, and less than five years in blue.

The surveys closest to this paper, in the top right corner of Fig. 1, covered a wide range of MAS decision making methods including game theory, reinforcement learning (RL), swarm intelligence, and evolutionary computing, and discussed multiple applications including robot soccer, prey-predator pursuit, air traffic control, and others [4], [5]. However, these surveys have become outdated and do not cover some key advancements in the field such as the contributions of deep learning. Some surveys covered a wide range of models but focused on multiple problems in robotics. Seven main research areas on multirobot systems (MRSs) were discussed in [6], including robot architectures, mapping and exploration, motion coordination, and object transport and manipulation. Multiple unanswered research questions were identified in [6] including complex task automation using MRS. Cooperative control of multivehicle systems and their applications in the military, transportation systems, and mobile sensor networks were surveyed in [7] who concluded that additional work in system integration, distributed embedded system verification and decision making at higher level abstractions was necessary before successful deployment of MRS. Yan *et al.* [8] surveyed multiple aspects of robot coordination including decision making, planning, and communication, and observed that more powerful coordination schemes are necessary to automate complex tasks.

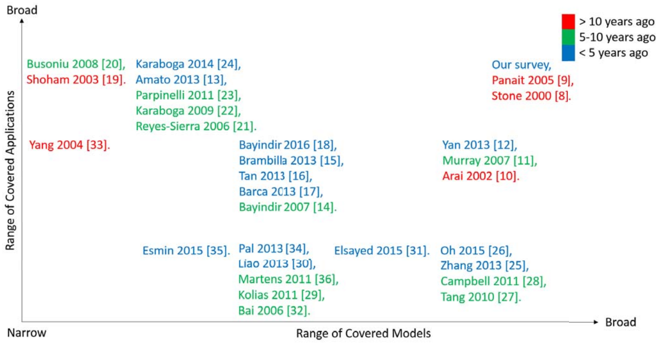

Fig. 1. Scope of existing surveys on MAS decision making.

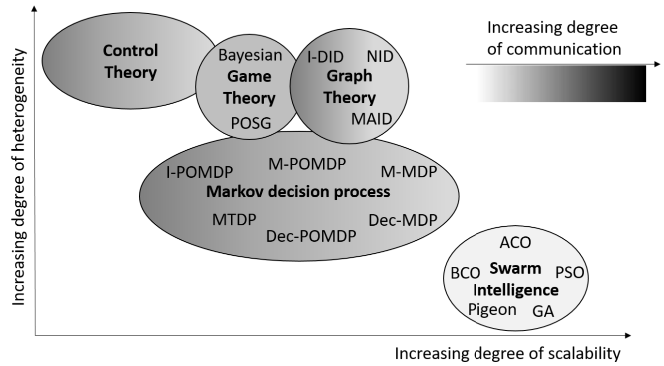

Fig. 2. Comparison of decision making model frameworks.

Surveys discussing one model and its applications included recent work on decentralized partially observable Markov decision process (Dec-POMDP) [9], swarm intelligence in robotics [10]–[14], and multiagent RL [15], [16]. Multiobjective particle swarm optimization (PSO) variants [17], algorithms based on bees [18], metaheuristic algorithms [19], and artificial bee colony variants and applications [20] have also been surveyed.

The following surveys discussed multiple models for one application such as formation control and coordination [21], [22], task allocation [23], [24], intrusion detection [25]–[27], and smart electric grids [28]. Finally, certain surveys focused on a single model applied to one application such as multiagent RL for robotics [29], swarm intelligence for robot path planning [30], PSO for clustering [31], and swarm intelligence for data mining [32].

In this paper, we survey existing cooperative MAS decision making models including Markov decision process (MDP) and its variants, game theory, and swarm intelligence. Fig. 2 depicts a fuzzy comparison between the discussed models based on three criteria: 1) heterogeneity; 2) scalability; and 3) communication bandwidth. While other models exist, such as belief-desire-intention models based on the human's practical reasoning theory [33] and independent choice logic which combined probabilistic information with logic programming to represent knowledge [34], their MAS extensions [35], [36] have not been widely adopted in the recent work. Multiple methods that find optimal or suboptimal action sequences for the various decision making models are surveyed. These include RL, dynamic programming (DP), recurrent neural networks (RNNs), and evolutionary computing, to name a few. We present decision making applications in robotics, wireless sensor networks (WSNs), traffic signal control, and others. Finally, we discuss some of the remaining challenges in cooperative MAS decision making such as leveraging big data advancements, creating more scalable, distributed and computationally efficient algorithms that can tackle more complex tasks, and developing evaluation standards.

In what follows, we first define key terms in the field of MAS in Section II. Then, we introduce various decision making models in Sections III–VI and their applications in Section VIII. Remaining challenges and insights on future research directions are discussed in Section IX before concluding in Section X.

II. Multiagent Systems

In this section, we define and categorize MAS before discussing its constituting blocks: agents and local interactions. Then, we focus on key terminology for decision making problems, which is an element of intelligent agents.

A. *Multiagent Systems*

MAS are composed of multiple autonomous, interacting agents that have common or conflicting goals and sensory information [37]. MAS are generally decentralized, asynchronous systems but can sometimes be centralized or hybrid. Their evaluation criteria include domain specific performance metrics and domain invariant criteria such as time and space complexity, load balancing, fairness, resource utilization, communication overhead, robustness, and scalability [38]. MAS have been categorized based on multiple criteria such as diversity of agents, communication capabilities, and interaction types. Agent heterogeneity stems from diverse sensing and actuating capabilities, computing resources, cognitive algorithms, and morphology [39]. Considering agent interaction complexity leads to three classes of MAS: 1) no direct interaction; 2) simple interaction; and 3) complex conditional interaction [40].

B. *Agents*

An intelligent agent is an autonomous entity capable of performing actions on its environment and perceiving its environment, aiming to accomplish a goal [41]. It can be a physical entity such as robots with sensors and actuators or a virtual entity such as software agents. An intelligent agent exhibits the fundamental properties of perception, reasoning, learning, decision making, problem solving, interaction, and communication [42]. It is evaluated based on its solution optimality, generality, robustness, efficiency, autonomy, and ability to learn and improve [42]. Agents are categorized based on many different criteria. One categorization depends on the decision making algorithm's instigator and results in three types of agents: 1) reactive; 2) deliberative; and 3) hybrid. Reactive agents react to environmental changes. Deliberative agents initiate actions without external triggers. Hybrid agents can react to the environment or initiate actions based on their planning algorithm. Another categorization, proposed in [41], is based on the agent's underlying architecture and contains four classes: 1) simple reflex agents; 2) model-based reflex agents; 3) goal-based agents; and 4) utility-based agents. Simple reflex agents react to current sensory input only while model-based reflex agents keep an internal state of the environment. Goal-based agents perform actions that lead to accomplishing their goals and utility-based agents maximize their utility.

C. Interactions

In addition to the complexity of interactions, MAS can exhibit different types of interactions based on agent goals, resources, and skills [43], [44]. Broadly speaking, interactions can be positive or negative. In the former, agents aid each other in accomplishing their goals, while in the latter, agents actively impede other agents' progress. Positive interaction can be further divided to collective, cooperative, collaborative, and coordinative. In collective interaction, agents are unaware of other agents' existence but share a common goal and each agent contributes to its completion, as in robot formation control and foraging. Cooperative interaction is similar to collective interaction except that agents are aware of other agents' existence. Examples include search and rescue, exploration, and object displacement. In collaborative interaction, agents do not have common goals but help each other accomplish their individual goals. Finally, in coordinative interaction, agents within an environment work together to minimize interference and complete their individual goals; MRS path planning is one example. Negative interaction can be either conflicting where agents do not have enough resources to complete their goals and fight for external resources or competitive where agents have conflicting goals. In this paper, we focus on cooperative MAS since it is an integral part of many smart city systems but still has many open research questions before effective deployment in real-world scenarios is possible. It is considered by some one of the more challenging interactions due to the need for high correlation and synchronization between agents and time sensitivity of agents' actions, especially in robotics. However, some of the models discusses in this survey can be applied to MAS with positive interactions such as swarm intelligence in collective MAS, game theory in collaborative MAS, and graph theory in coordinative MAS.

D. Decision Making

Decision making, or planning and control, enables an agent to accomplish its goals by determining what action to perform. The decision making problem can either be episodic or sequential [41]. The output of the former is a single action while the latter produces a sequence of actions or policy. The decision making algorithm is evaluated based on policy optimality, search completeness, time complexity, and space complexity. A policy is optimal if it has the highest utility. A search algorithm is complete if it guarantees to return an optimal policy in finite time, when it exists. Time complexity quantifies the amount of time needed to search for a solution while space complexity quantifies the amount of computational memory needed. In this paper, we focus on sequential decision problems which can be of two types: 1) finite or 2) infinite horizon. Finite horizon implies that decisions need to be made for a finite number of time steps while infinite horizon problems last forever. When discussing decision making in the context of MAS, learning can be either centralized or decentralized. Reference [5] used the terms team learning and concurrent learning. In the former, one learner learns policies for all agents in the system while in the latter, each agent learns its own policies in parallel to other agents. Credit assignment, how to distribute rewards among cooperating agents, is one problem that arises and should be appropriately handled to achieve optimal performance. Communication, whether direct or indirect, is another issue in cooperative decision making that should be considered.

III. Markov Decision Processes

In this section, we present the MDP formulation, its extension to MAS and partially observable environments, and conclude with some insights on this method.

A. Markov Decision Process

An MDP, a discrete time stochastic control process, is characterized by fully observable states and outcomes that are influenced by decision makers. It satisfies the Markov property which states that decisions made at the current time step rely on a finite number of previous time steps. It can also be viewed as a fully observable stochastic game with a single player. In some texts, it is referred to as a dynamic program, stochastic dynamic program, sequential decision process, and stochastic control problem. An MDP is defined by the tuple (S, A, P, R, γ). S represents the set of states, s, of the environment. A represents the set of actions, a, an agent can perform. In some states, certain actions are not permissible, i.e., only a subset of the actions can be performed, denoted by A_s. P represents the transition probability. $P_a(s_i, s_j)$ denotes the probability that the environment will transition to state s_j from state s_i when an agent performs action a. R represents the reward. $R_a(s_i, s_j)$ denotes the received reward when performing action a and the environment goes from state s_i to state s_j. γ represents a discount factor, $\gamma \in [0, 1)$, that gives more weight to present reward than future reward. MDP was found to be P-complete [9]. Constrained MDP impose additional constraints on MDP, resulting in more than one cost for every action and the final policy depends on the initial state of the process [45]. Time-dependent MDP [46] extends MDP to continuous time state spaces where value iteration is performed on a piece-wise linear value function.

A solution to an MDP is a policy that should be performed by an agent to maximize its total reward, measured using a value function V. A policy function maps states to actions: $\pi : s \rightarrow a$ or $a = \pi(s)$. Action selection methods determine what action to select next, based on the estimated value functions of the action set, while considering the exploration-exploitation trade-off. A greedy method picks the action with the highest value, ϵ-greedy selects the best action with a probability of $1 - \epsilon$. Boltzmann exploration assigns probabilities of selecting actions using an exponential function of the value function.

Many algorithms have been proposed to find optimal and suboptimal policies for MDP. DP [47], temporal difference learning [48]–[50], policy search [51], and linear programming [52], [53] require models of the state transition and reward functions. If these models are unknown or too complex, approximate methods are adopted and include model-free RL approaches like Q-learning [54] and SARSA [55], evolutionary computing [56], [57], RNN [58]–[63], and deep RL [64], [65]. Distributed optimization methods have been adopted to solve MDP problems. First, the alternating direction method of multipliers decomposes the MDP into subproblems. Then, a distributed Newton method [66] or linear programming algorithm [67] find the optimal policy.

MDP extensions have been proposed for MAS. Multiagent MDP (M-MDP) extends MDP to MAS by assuming a joint action space with a team reward model and fully observable environment. A central learner learns a vector of actions that should be performed by the agents and the reward is common to all agents [37]. The worst-case complexity of finite horizon M-MDP is P-complete [68] which is solvable in polynomial time by a Turing machine, an abstract model of computing devices. As the number of agents increases, the joint state and action spaces' dimensionalities increase exponentially. To ease the computational burden, independence is assumed to make objective functions factorable. Solving the problem iteratively also reduces the computational complexity. Distributed implementations of the central learner have been developed for factorable objective functions [69]. On the other hand, decentralized MDP (dec-MDP) assumes an independent action space with local reward and jointly fully observable environments [37]. In other words, individual agents view a partially observable environment but the aggregate observations of all agents in the MAS make the environment fully observable. Finite horizon dec-MDP was proven to be worst case NEXP-complete (solvable in exponential time using a nondeterministic Turning machine), when three or more agents are considered [70]. Since actions and rewards are local, this approach falls under the concurrent learning class of MAS learning. Assuming agent observations and transitions are independent, the model is known as TI dec-MDP and its complexity is NP-complete, meaning a solution can be found in polynomial time by a nondeterministic Turing machine. This model can be further simplified by assuming independent rewards to obtain a P-complete complexity in the worst case.

B. Partially Observable MDP

Partially observable MDP (POMDP) is a generalization of MDP to partially observable environments and is defined by $(S, A, P, \Omega, O, R, \gamma)$, where Ω represents the set of observations, O is the observation function, and the remaining terms are as defined for MDP. POMDP was found to be PSPACE-complete [9]. Many algorithms have been proposed in the literature to provide exact and approximate solutions for the POMDP and its variants, including value iteration [71], [72], expectation maximization [73], [74], nonlinear optimization [75], quadratically constrained linear programming [76], Monte Carlo methods [77], [78], and DP [79], [80] when state and transition models are known. When they are not known, heuristic search algorithms [81], genetic algorithms [82], RNN [83]–[86], and model-free RL methods were applied. Liu *et al.* [87] learned the number of states to represent in a nonparametric scheme and used RL to find policies for POMDP. Unsupervised learning was adopted to learn an observation space transformation to a latent representation space where policies are learned, in [88]. Forward simulation was used to estimate policy utilities [89].

Dec-POMDP generalizes POMDP to MAS where rewards are common and based on joint actions but observations are individualistic [9]. The goal is to maximize the reward of the entire system as agents collaborate to achieve a common task. Communication among agents can be explicit (Dec-POMDP-COM) or implicit (Dec-POMDP). This model is NEXP-complete [70]. Approximate solutions have been proposed based on bounded policy iteration [90], Q-value function methods [91], multiagent A^* [81], genetic algorithms [82], DP [92]–[94], and a Bayesian learning, stick-breaking policy algorithm [95]. A set of approximate inferences and heuristics including bootstrapping were used to find approximate solutions to Dec-POMDP in [96].

The multiagent team decision problem [97], equivalent to Dec-POMDP when agents have perfect recall [98], extends economic team theory to robotics. It includes models for implicit and explicit communication and is proven to be NEXP-complete. Multiagent POMDP extends M-MDP to partially observable environments, and is PSAPCE-complete which means the algorithm's memory requirements are polynomial function of the input size. Like M-MDP, it is a team learning approach that has a central learner, and employed Bayesian RL framework to learn policies [99].

Networked distributed POMDP (ND-POMDP) assumes local interaction among agents to reduce the computational cost of finding policies [100]. ND-POMDP is a factored Dec-POMDP model where observations and transitions are independent and rewards are divided among neighboring agents. Its worst case computational complexity is NEXP-complete. Algorithms used to find policies for this model include multiagent RL [101], DP [102], and distributed constrained optimization [100]. Interactive POMDP (I-POMDP), a concurrent learning approach, generalizes POMDP to MAS by modeling other agents in the system while maintaining a belief of the system state [103]. Finitely nested I-POMDP is PSPACE-complete [98] and approximate solutions have been proposed based on particle filters [104], value iteration [105], and Monte Carlo sampling methods [106].

C. Some Insights

MDP and its variants have been widely adopted in many complex MAS decision making problems, despite the very restrictive Markovian assumption. Even though these models do not scale well, they are able to handle agent heterogeneity. Recently, deep learning approaches have been adopted to solve various MDP models and provided a roadmap to solve non-Markovian models as well. In addition, deep learning has allowed the extension of MDPs from the discrete space to the continuous space, which is more suitable for robotic MAS.

IV. Game Theory

Game theory develops models of interaction between rational decision makers under different circumstances [107]. It has been applied in many fields from economics and psychology to artificial intelligence. In this section, we focus on two types of games that have been commonly applied to cooperative MAS in artificial intelligence: 1) stochastic games and 2) Bayesian games.

A. Partially Observable Stochastic Games

Stochastic or Markov games [108] are sequential probabilistic games. They can also be viewed as a generalization of repeated games where a game from a collection of normal form games can be played at a given step [37]. Payoffs depend on both actions and the state of the game at the current stage. Players' actions and the game's current state cause the game to transition to other states. Stochastic games are represented using the tuple (Q, N, A, P, r), where Q denotes the set of games that can be played, N denotes the set of players or agents participating in the game, $A = A_1 \times \cdots \times A_N$ denotes the actions of the players, P denotes the transition probability function, and r denotes the reward or payoff. They belong to the complexity class $NP \cap co - NP$ [109] and can be solved using DP [108], Q-learning [110], and linear programming under certain conditions [37].

To solve a game, a strategy profile or solution concept must be obtained; it is a strategy for each player. A strategy, equivalent to a policy in MDP [41], is a rule used by agents to select an action. An equilibrium strategy is defined as the best response of an agent to another agent's strategy, i.e., the agent cannot improve its expected utility by changing its strategy. It does not always exist in stochastic games but may exist under restricted conditions. For example, stochastic games with a finite number of players, actions and states always have a Nash equilibrium, defined as the strategy profile which maximizes each player's utility knowing the strategy of others in the game [111]. Evolutionary stable strategy is a refinement of the Nash equilibrium which requires a strategy to be stable to any perturbations that may occur to the games as they evolve [112] and was extended to stochastic games [113]. Stochastic games have been shown to have an evolutionary stable strategy under certain conditions [114].

Partially observable stochastic games (POSGs) extend stochastic games to partially observable environments where the payoffs are not known to the players. They are represented by the tuple (Q, N, O, A, P, r, b^0), where O denotes the set of observations and b^0 denotes the initial state distribution. POSG have been used to model learning sequential decision making in cooperative MAS [115]. Finding a Nash equilibrium for POSG belongs to the NP-hard computational complexity class [116], meaning they are computationally at least as difficult as NP problems which are solvable by a nondeterministic Turing machine in polynomial time. POSG subclasses include MDP, POMDP, and their MAS extensions.

Many exact and approximate solutions have been proposed for POSG. An iterative method to eliminate dominant strategies was proposed in [116]. Hansen *et al.* [93] combined a generalized version of the DP used for POMDP and eliminated dominated strategies to find a solution to POSG. When agents use the same payoffs, this approach can converge to an optimal solution. The proposed method was tested on multi access broadcast channel control and compared to a policy tree building brute force algorithm. POSG have also been used to model cooperative MAS decision making in partially observable Markovian environments [115], [117]. However, this model's solution is intractable as the number of agents increases. Therefore, an approximate solution was computed based on Bayesian games to achieve decentralized control in robot teams with limited communication. The algorithm was validated on the two-robot tag problem, two-agent lady, and tiger problem and multiple access broadcast channel problems.

B. Bayesian Games

Bayesian games are games with incomplete information. Generally, these uncertainties can be modeled as uncertainties in agents' payoffs [37]. They are defined by (N, G, P, I), where N represents the set of agents, G represents the set of games the agents might be playing, P represents the common prior distribution over all the games, and $I = (I_1, \ldots, I_N)$ represents the partitions of G, for each agent. Examples of Bayesian games include signaling games, bargaining, auctions, and market competitions. Strategic policies can be obtained by converting incomplete games to imperfect information ones. Solving for the Bayesian Nash equilibrium, the Nash equilibrium in Bayesian games, includes best response, RL or other learning rules, linear programming [118], and Monte Carlo methods [119]. Bayesian Nash equilibrium, which consists of a strategy profile and a player's belief about other players' types, always exists.

C. Some Insights

While game theoretic approaches had been mainly used in competitive MAS, some models have gained popularity in cooperative MAS due to the agents' capabilities of modeling other agents in the game. This property can be useful in robotic systems where robots are unable to communicate with others. However, this restricts the number of agents in the system due to increasing computational costs. Game theory's systematic mathematical approach has been an attractive quality for many applications but combining it with some heuristic approaches such as deep learning might lead to improved performance in robotic applications and others.

V. Swarm Intelligence

Swarm intelligence describes the behavior of decentralized cooperative agents, whether natural or artificial, working toward a common global goal [120]. Self-organized and distributed behavior of locally aware and locally interacting agents are pillars of swarm intelligence [121]. Systems modeled in this fashion generally consist of many autonomous but homogeneous agents implementing simple rules with agent interactions restricted to local neighborhoods.

A. Biologically Inspired Algorithms

Swarm intelligence was inspired by many social insects and animals including ants, bees, wasps, termites, bats, fish, and birds. In some ways, swarm intelligence is similar to RL; both are iterative algorithms that use a reinforcement signal to learn a solution [121]. However, the reinforcement signal modifies the behavior of the agent differently in both algorithms.

Many algorithms have been inspired by bee colony behavior. Bee colony optimization [122] is based on direct communication among agents performing a series of moves for a certain duration based on the strength or fitness of the solution, also known as "waggle dancing." This recruits other agents to the most fit solution. Navigation is based on path integration where agents continuously update a vector indicating the position of the start location. Ant colony optimization (ACO), inspired by ant colony behavior, is a class of algorithms that rely on indirect communication [123]. Navigation is based on depositing pheromones along the trail. A more fit solution results in stronger pheromones on the trail that lead to recruiting more agents. PSO is inspired by flocks of bird and schools of fish [124]. Agents navigate the environment searching for better solutions using principles from birds' movements. A pigeon inspired optimization algorithm relied on the magnetic field, sun and landmarks to achieve path planning [125]. Distributed implementations of ACO [126], [127], and PSO [128] have been developed to speedup convergence.

B. Some Insights

While such systems exhibit desirable properties like robustness, flexibility, scalability, low complexity, inherent parallelism, and fault tolerance [11], [129], they have important limitations. Most swarm systems consist of identical agents, leading to their limitations according to [129]. The agents must be homogeneous or can be divided into a small number of homogeneous clusters following simple rules to make decisions. However, there are many applications, such as search and rescue operations, that require heterogeneous, complex agents working toward a common goal.

VI. Graph Theory

Decision making in MAS have been modeled as graphs with nodes representing agents and edges representing interactions and information flow among agents [130]. In this section, we focus on one popular approach called influence diagrams (IDs), briefly discussing the model and some of its strengths and weaknesses.

A. Influence Diagrams

IDs are referred to as decision networks in [41] and are a graph theoretic approach that provide a framework for decision making by adding actions and utilities to Bayesian networks [131]. Chance nodes (ellipses) represent random variables. Decision nodes (rectangles) represent choices available to the agent and utility nodes (diamonds) compute the utility of these choices. The action with the highest utility is chosen. IDs can be converted to decision trees by traversing the diagram from top to bottom, creating a node in the decision tree when a decision node is encountered and adding edges with values equal to probabilities of parent nodes; leaves portray the utility of a path. IDs require the optimization of all parent nodes of a decision variable [34]. Dynamic IDs (DIDs) extend IDs to sequential decision making problems by combining DP with IDs [132] and have been viewed as computationally equivalent to POMDP [133]. They exploit the separability of the value function to generate computationally efficient solutions.

Multiagent IDs (MAIDs) generalized IDs to MAS by generating decision rules that depend on decision rules made by other agents [134]. This is graphically represented by connecting decision nodes that depend on each other; a directed relevance graph is thus produced. MAIDs represent games with imperfect information graphically and are an alternative to the normal and extensive forms of game representation [133]. They can be converted either to extensive form games or to IDs and then solved.

A network of IDs (NIDs) is built on top of MAIDs to account for uncertainties in other agents' decision making and hierarchy of beliefs [135]. This formalism can represent irrational behavior and distinguishes between different agent models in the systems, i.e., it does not treat all other agents identically. Acyclic NIDs can be solved using a bottom up approach by converting each block to a MAID and solving it. Duplicates are included to account for beliefs about others' strategies. Cyclic NIDs are converted to acyclic NIDs and solved. However, both MAID and NID are applicable to episodic decision making only. Interactive DIDs were proposed in [133] as an MAS extension of DIDs and can be viewed as computational counterparts of I-POMDP. Models of other agents are clustered to reduce computational complexity but lead to approximate solutions.

B. Some Insights

Graph theory models the interaction of agents, allowing them to exchange information and make decision accordingly. However, the computational complexity of this approach increases exponentially in densely connected graphs with many nodes (agents). The main benefits of graph theory in MAS come from combining it with other approaches such as MDPs and control theory (discussed next) to extend these approaches to MAS. Furthermore, exploiting special structures such as sparsely connected dense subgraphs are a common approach to reduce computational cost and improve performance.

VII. Control Theory

Control theory is an established field that aims to control physical systems by designing controllers using modify the input to achieve the desirable output. Its subfields include nonlinear, adaptive, optimal, robust, and stochastic control, to name a few, and produce controllers with various properties to overcome limitations imposed by the real-world environment they operate in. However, as automation problems became more complex, researchers extended control theory to MAS by developing distributed controllers. We present a brief overview of this broad field next.

A. Distributed Cooperative Control

Distributed controllers are designed by combining concepts from control and graph theory. Specifically, interactions among agents are modeled using graph theory and the control problem is decomposed among the agents to obtain a distributed controller. The amount of communication among agents is dependent on the design of the distributed controller and can vary based on the nature and complexity of the task (whether it is easy decomposable), the optimality of the control algorithm and other factors. Since many controllers are based on optimization algorithms, distributed optimization is an integral part of distributed control [136]. Unlike other approaches, distributed cooperative controllers designed using control and graph theory can be mathematical validated to prove optimality, stability, robustness, and convergence, to name a few properties.

Distributed controllers have been applied to various control problems. For example, a Lyapunov-based voltage and frequency controller was designed for micro-grid systems that only requires local communication among neighbors [137] and a secondary voltage distributed controller based on input–output feedback linearization that requires sparse communication [138]. A Lyapunov-based distributed lead-follower control system was developed that scaled to large MAS when the interaction topology is an undirected graph [139]. Distributed consensus tracking was achieved by designing: distributed adaptive controllers in weakly connected, directed graphs [140], a distributed optimal control algorithm [141], and nonlinear distributed impulsive control (control signals are given as impulses instead of continuously) with delayed impulses in undirected graphs [142]. Stochastic sampling in leader-follower consensus problems has been shown to improve scalability of MAS [143]. Distributed impulsive control has also been applied to heterogeneous MAS synchronization problems [144]. Yang *et al.* [145] proposed distributed output regularization using adaptive control in MAS with a switching topology. Other applications include formation control [146] and navigation [147] in MRS.

B. Some Insights

While control theory adopts systematic mathematical approaches to develop controllers, some systems are simply too complex and intractable for such methods. For example, most algorithms assume linear systems. Therefore, data-driven methods such as those in distributed artificial intelligence are necessary to automate certain complex tasks in real-world environments. Nevertheless, distributed cooperative controllers are necessary in some applications where sufficient data is not available or mathematically proven optimal controllers are crucial like in aviation or military domains.

VIII. Applications

MAS decision making models have been applied to many problems in various fields from robotics to WSNs. Next, we mention some of the problems that have been solved using the aforementioned decision making models.

A. Robotics

Cooperative MRS have been applied to many problems that require various degrees of coordination. Loose coordination examples include formation control and foraging, while tight coordination examples include object transport and robot soccer. Environment uncertainty, robot actuating and sensing diversity, system scalability, real-time processing, and limited computational resources are a few challenges that should be addressed when designing decision making algorithms. Decisions related to robot actions, information sharing, and coalition formation, are essential to the successful deployment of robots in real world environments. POSG has been applied to multiple problems in robotics [148], [149]. MDPs [150] and POMDP [151]–[155] have been used for robotics coordination including robot soccer [156]. Graphical models for consensus [157], formation [158], [159], and rendezvous [158] have also been investigated. Finally, swarm intelligence has been applied to underwater environments [160], 3-D space [129], and robot path planning problems [30], [125]. It has been applied to dynamic task allocation [161], distributed localization problems [162], foraging tasks [163]–[165], collision free navigation [166], [167], and communication free flocking with minimal memory requirements [168]. Swarm-bots, wheeled robots that can physically connect to each other and form larger entities, accomplished coordinated motion, self-assembly, cooperative transport, goal search, and path formation [169], [170]. The thermotactic behavior of honeybees inspired the decision making of a swarm of microbots with limited communication capabilities in spatial behavior problems [171]. Swarmanoids, a heterogeneous system composed of three types of complementary swarm robots, performed complex tasks like object retrieval in 3-D space [129].

B. Repeated Coalition Formation

Forming groups of agents that change based on environmental conditions is critical to the successful deployment of MAS in real-world environments. Repeated coalition formation under uncertainty deals with forming time varying coalitions where agents do not have complete information about other agents' capabilities. Adopting traditional coalition formation methods such as auctioning and search algorithms cannot handle uncertainty since they assume complete knowledge of agent capabilities. Therefore, this problem has been modeled as a sequential decision making problem by many researchers and solved using some of the decision making models discussed in this paper, which can handle information uncertainty. This allows the dynamic formation of robot teams where uncertainty is high and can lead to the automation of complex tasks that was previously unfeasible.

While searching for a solution that strikes a balance between redundancy for fault tolerance and agent's skill complementarity is challenging enough, attempting to do so with incomplete and noisy information about agents' skills further complicates matters. Dynamically reforming coalitions also poses its own challenges by requiring the algorithm to determine the lifetime of a coalition. However, repeated coalition formation with uncertainty allows MAS to cope with the stochastic environments and complex tasks.

Matthews *et al*. [172] assumed the problem was fully observable and adopted MDP to model a football team formation problem. Agent transitions between coalitions were modeled as an MDP, the Shapley value and marginal contributions were used to prune the search space and the best coalition structure was found using Markov probability distributions [173]. A POMDP model was also adopted to allow agents to learn other agents' capabilities by interacting with each other [174]. IDs solved the problem of coalition formation for complex real-world missions by selecting a subset of coalition formation algorithms suitable for the problem at hand [175]. Swarm intelligence was used to search for the best coalitions to form [176]. Coalition games were generalized to problems with incomplete information through Bayesian games [177], [178].

C. Intelligent Transport Networks

With the increased awareness on sustainable living, transportation systems are challenged to endorse cutting edge technology and provide better services, while keeping an eye on safety and greener emissions. Intelligent transport networks are formed of autonomous or semi-autonomous communicating vehicles and road infrastructure such as traffic signals and road sensors. Decisions such as when to close a road or change a traffic light color to reduce traffic congestion, give directions to emergency response vehicles to avoid congested roads, improve road safety based on weather conditions, and others, are critical to make transportation smarter. To run efficiently, decisions need to be made in real-time on devices with limited computational resources. POSG modeled directional routing and scheduling of packet delivery in vehicular ad hoc networks [179]. POMDP was used to perform automated driving in urban traffic while dealing with sensor uncertainties [180]. Multiagent RL has been used for routing algorithms [181]–[183], adaptive broadcasting [184], adaptive data collection [185], and traffic signal control [186]–[193]. Intelligent transportation systems have utilized swarm intelligence to control traffic light scheduling [194], model complex transportation systems [195], and develop routing protocols for vehicles [196], [197] and for information dissemination [198]–[202].

D. Wireless Sensor Networks

WSN are a collection of autonomous computing and sensing devices with limited computational resources. Their presence is ever increasing with the decreasing cost and size of hardware and emergence of Internet of Things. Integrating decision making in these networks allows us to implement functionality beyond simple information retrieval, making the integration of WSN with other smart city MAS, such as autonomous vehicles, electric grids, and transport networks, feasible. The application of MDP was surveyed to model various problems in WSN including intrusion detection, sensor coverage, object detection, data exchange, topology formulation, and other problems [203]. POMDP have been used for performance optimization [204], data and memory access control [205], and sleep scheduling [206]. IDs were used for lighting control in WSN and provided robustness to sensor uncertainties [207]. Swarm intelligence has been used for routing in WSN [208]–[214], clustering [215], cluster head selection [216], for security protocols [215], [217], and node positioning and localization [218], [219].

E. Intrusion Detection

An essential component of network security is detecting threats before they can compromise the network. Since networks are inherently decentralized, detecting threats can be modeled as an MAS decision-making problem where agents cooperatively determine whether a threat is present. Intrusion detection systems monitor activities in network infrastructures such as WSN and mobile ad hoc networks to identify malicious behavior. It involves detecting malicious packets, tracking their sources and optimizing performance of networks. These systems are considered MAS because each node on the network contributes to keeping the network secure, by making decisions related to the maliciousness of packets. Intrusion detection is a difficult problem because running the decision making algorithms should not use up a significant portion of the network nodes' limited computational resources while identifying threats as early as possible on a wide variety of network technologies.

Many models have been adopted in intrusion detection systems, as surveyed in [220]. MDP identified the network's most vulnerable nodes based on attackers' previous behaviors [221], [222]. Bayesian games modeled attacker/defender games [223]. Multiagent RL were used to implement distributed intrusion detection systems [224]. Swarm intelligence is a popular approach to intrusion detection, evidenced by the recently published surveys [25]–[27]. PSO has been widely applied in combination with support vector machines [225]–[227] and linear programming [228]. ACO was used in IP traceback problems [229].

F. Other Applications

Cooperative MAS has been applied to many other fields. Noteworthy applications, briefly discussed next, are cognitive radios, smart electric grids, resource allocation, and distributed optimization. Traditional radio paradigms suffer from spectrum scarcity and usage inefficiency. Cognitive radios have been presented as one possible solution. A cognitive radio is a smart radio that efficiently utilizes the available spectrum. MDP and RL were used to model and solve spectrum sensing and management problem [230]–[234]. Jamming in networks were modeled using stochastic games [235] and spectrum sharing was modeled using game theory [236]. Decision making models to cognitive radios reduces the wasted, already scarce, spectrum resources and improves their efficiency in switching frequencies.

Smart electric grids will be the primary method of power distribution in smart cities where efficient scheduling, generation and distribution of power are essential. However, the unpredictability of demand and supply as well as plant diversity and plant failures are some of the challenges faced in this field. POSG [237], POMDP [238], [239], multiagent RL [240], [241], and swarm intelligence [28], [242]–[244] have been adopted to model and solve power distribution, scheduling, power flow, and load forecasting problems.

Resource allocation aims to distribute heterogeneous resources in a fair and efficient manner to maximize resource utilization. Resource allocation has many applications in resource constrained domains where many agents are battling to gain access to these scarce resources such as robotics and cloud computing. Decision making models adopted for this problem solve this problem more efficiently than other approaches such as search or constrained optimization methods. Adopted models include POMDP, used to minimize network bandwidth congestion and fairly allocate resource [245], Bayesian games [246] and stochastic games with multiagent RL used for job, and resource scheduling in grid computing [247].

Distributed optimization, a useful tool in many fields including robotics, electric grids, and large-scale optimization problems, consists of optimizing an objective function in a distributed fashion. MAS decision making leveraged consensus and communication rules to model distributed optimization; each agent optimized part of the objective function before combining their results [248]. For example, game theory has formulated distributed optimization problems as games [249], [250]. In [251], potential games and cooperative control provided a theoretical framework to formulate distributed optimization problems. Also, swarm intelligence including PSO [252], [253] and an algorithm that mimics bacterial foraging [254] have been used to solve multiobjective optimization problems. Graph theory including time-varying directed graphs [255] and weight-balanced directed graphs [256] have modeled information exchange in a distributed optimization framework.

IX. Challenges

Although MAS decision making has seen significant improvements in the past decade, it is still plagued with many issues. To reap all the benefits of the Internet of Things boom and improve smart cities and smart living, decision making systems need to address some of the remaining challenges.

A. Scalability

Decision making algorithms should be scalable, especially in heterogeneous MAS, to accomplish more complex tasks. The scalability of current models greatly relies on agent homogeneity and the level of interaction. Swarm intelligence can scale to large MAS since agents are homogeneous and interaction is minimal and restricted to the agent's neighborhood. MDP variants and game theoretic models do not scale well since the complexity of the algorithm increases exponentially due to the model formulation that results in exponentially large state spaces. Using the graph theoretic formulation for large MAS results in densely connected graphs which are computationally expensive.

B. Computational Complexity

Decision making algorithms should be computationally efficient due to the need for real-time decision making in some applications or the lack of enough computational resources of agents. Robots generally have limited on board computational resources due to size and weight constraints and might not be able to offload their computations to the cloud due to bandwidth scarcity, poor or unreliable connectivity, and minimum latency requirements. Agent interactions MAS increases the computational cost per agent as the number of agents increases, especially in methods that extend single agent models to MAS if careful consideration of interaction cost is not performed. Tightly coordinated tasks also increase the computational burden due to the large amount of communication and data exchange among agents. Decision making algorithms should be designed with all these constraints in mind to successfully complete complex tasks.

C. Dynamic Environments

The environment's dynamic and unpredictable nature makes it difficult to foresee, design and test an agent that can handle all these situations. Therefore, decision making algorithms should generalize well to situations that have not been learned or tested. They should be able to adapt to the dynamic environment and various uncertainties it might encounter and should be robust to noisy and incomplete information generated by sensors, and nondeterministic actions. POMDP, IDs, POSG, and Bayesian games are better suited to handle uncertainties than MDP and its variants that assume fully observable environments, because they account for partially observable environments, incomplete and imperfect information in their algorithms. Agent failures are also a source of uncertainty in MAS that hinder the completion of tasks. Unlike other models, swarm intelligence models are better suited to handle agent failures due to the homogeneous nature of agents and minimal interaction necessary. However, this is still an issue that needs to be considered whenever MAS are designed.

D. System Heterogeneity

Heterogeneous MAS can deal with environment diversity and complex tasks. However, this heterogeneity makes cooperative decision making more complex: agents need to model other agents when capability uncertainty exists, agent capabilities should be compatible, and agents should have a common language to communicate and interact, in addition to other issues. Swarm intelligence simplifies modeling by assuming all agents are homogeneous. Graph theoretic models, POSG and its subclasses can handle heterogeneous MAS if the state and observation spaces are designed appropriately. I-POMDP and I-DID inherently model other agents, making them better than other graph and game theoretic models in dealing with MAS heterogeneity.

E. Big Data

Recent advancements in processing big data has led to significant improvements in research areas like object recognition, speech recognition and natural language processing. The next step is to use this information to make better decisions in MAS and handle more complex tasks. Decision making has yet to maximize its benefits from big data. Algorithms that model and generate representations of such data like convolutional neural networks (deep learning) produce computationally expensive models that are not suitable for computationally limited agents or decision making algorithms whose computational cost grows exponentially with the dimensionality of the data. Yet, allowing agents to access these models through the cloud has its own complications with respect to cloud accessibility, bandwidth constraints, representation compatibility, privacy, and security.

F. Evaluation Standards

Evaluation standards are necessary in MAS decision making to compare proposed algorithms and assess the state-of-the-art. General metrics include solution optimality, algorithm completeness, and algorithm time and space complexity. However, additional evaluation metrics of MAS decision making are necessary to enable better comparisons. Some work has developed evaluation metrics and workflows to quantify the performance of MAS. Braubach *et al.* [257] developed abstract metrics that would be specialized for MAS applications, and include function (e.g., restrictions), usability (e.g., simplicity), operating ability (e.g., performance), and pragmatic metrics (e.g., installation). Lass *et al.* [258] distinguished between two metric categories: 1) effectiveness (e.g., success, failure, and 90% accuracy) and 2) performance (e.g., resource consumption and time complexity), that could be applied to four MAS levels (agent, framework, platform, and host). They presented a framework to select appropriate metrics for a given application and performed a case study on a distributed constrained optimization problem. Di Bitonto *et al.* [259] developed a hierarchical metric system where both interagent (communication and cooperation) and intra-agent metrics measured environment complexity, agent rationality, autonomy, reactivity, and adaptability. This system was tested on a knowledge management problem for the automotive industry with two agents only. Marir *et al.* [260] proposed an evaluation platform that included metrics like average of communication load and validated the platform on an auctioning problem. Nevertheless, standards to evaluate and compare the performance of MAS on real-world environments are still underdeveloped. Existing metrics have been tested on a hand-full of narrow-scoped scenarios that did not necessarily include robot agents.

G. Other Challenges

Task complexity poses a challenge for decision making algorithms because they do not have the capability of recognizing what tasks can be decomposed into simpler tasks that they can complete. Adding this capability to decision making algorithms in MAS in addition to dynamically recognizing what tasks require tight coordination and what tasks can be accomplished with minimal interaction among agents will increase the scope of automated tasks. Learning algorithms for decision making and perceiving agents should be autonomous. Reducing the number of manually tunable hyper parameters that require human intervention will allow algorithms to generalize better to unknown environments.

X. Conclusion

This survey discusses decision making models and algorithms to find policies for cooperative MAS for different applications. MDP and game theoretic models, swarm intelligence, and IDs were covered, for which optimal and suboptimal policies were obtained using RL, DP, direct policy search, Monte Carlo methods, linear, quadratic and mixed integer programming, evolutionary computing, and RNN. MAS applications noted include smart electric grids, WSN, intelligent transportation systems, and robot teams performing search and rescue, object transport, and exploration and mapping. While state-of-the-art methods within the past five years are significantly better than their predecessors, research advances in this field are promising but still needs to answer many questions. Decision making algorithms should leverage big data advancements and the Internet of Things to obtain better policies, algorithms should be scalable as more complex tasks require larger MAS, and distributed algorithms should be adopted to ease the computational burden and run on computationally limited devices. Furthermore, evaluation standards or benchmarks need to be developed to enable comparison of algorithms and to facilitate their verification and validation. These improvements would take us a step closer to effective deployment of various MAS in smart cities.

Even though this survey focused on positively interacting MAS, MAS with negative interactions is has many real-world applications. Competitive MAS in robotics, intelligent transportation systems, smart electric grids, among others is an active area of research. Decision making models based on theories in economics, game theory, and psychology have been developed. MAS with conflicting interactions is also a prominent area of research especially in robotics and intelligent transportation systems where mobile vehicles use the same infrastructure and must co-exist with minimal interference to complete conflicting goals. Research areas such as conflict management, conflict resolution, and deceptive behavior modeling have emerged to address these issues.

References

[1] *Gartner Says 6.4 Billion Connected "Things" Will be in Use in 2016, Up 30 Percent From 2015*, Gartner, Stamford, CT, USA, 2015.

[2] *An Internet of Things That Will Number Ten Billions*, CNBC, Englewood Cliffs, NJ, USA, 2016.

[3] *First Workshop on Cognitive Architectures*, IBM, Armonk, NY, USA, 2015.

[4] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Auton. Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[5] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Auton. Agents Multi Agent Syst.*, vol. 11, no. 3, pp. 387–434, 2005.

[6] T. Arai, E. Pagello, and L. E. Parker, "Editorial: Advances in multi-robot systems," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 655–661, Oct. 2002.

[7] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *J. Dyn. Syst. Meas. Control*, vol. 129, no. 5, pp. 571–583, 2007.

[8] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 12, p. 399, 2013.

[9] C. Amato, G. Chowdhary, A. Geramifard, N. K. Ure, and M. J. Kochenderfer, "Decentralized control of partially observable Markov decision processes," in *Proc. IEEE 52nd Annu. Conf. Decis. Control*, Florence, Italy, 2013, pp. 2398–2405.

[10] L. Bayindir and E. Şahin, "A review of studies in swarm robotics," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 15, no. 2, pp. 115–147, 2007.

[11] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intell.*, vol. 7, no. 1, pp. 1–41, 2013.

[12] Y. Tan and Z.-Y. Zheng, "Research advance in swarm robotics," *Defence Technol.*, vol. 9, no. 1, pp. 18–39, 2013.

[13] J. C. Barca and Y. A. Sekercioglu, "Swarm robotics reviewed," *Robotica*, vol. 31, no. 3, pp. 345–359, 2013.

[14] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, Jan. 2016.

[15] Y. Shoham, R. Powers, and T. Grenager, "Multi-agent reinforcement learning: A critical survey," Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Rep. 06-025, 2003.

[16] L. Buoniu, R. Babuŝka, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[17] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 287–308, 2006.

[18] D. Karaboga and B. Akay, "A survey: Algorithms simulating bee swarm intelligence," *Artif. Intell. Rev.*, vol. 31, nos. 1–4, pp. 61–85, 2009.

[19] R. S. Parpinelli and H. S. Lopes, "New inspirations in swarm intelligence: A survey," *Int. J. Bio Inspired Comput.*, vol. 3, no. 1, pp. 1–16, 2011.

[20] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2014.

[21] Y. Zhang and H. Mehrjerdi, "A survey on multiple unmanned vehicles formation control and coordination: Normal and fault situations," in *Proc. IEEE Int. Conf. Unmanned Aircraft Syst.*, Atlanta, GA, USA, 2013, pp. 1087–1096.

[22] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015.

[23] S.-Y. Tang, Y.-F. Zhu, Q. Li, and Y.-L. Lei, "Survey of task allocation in multi agent systems," *Syst. Eng. Electron.*, vol. 32, no. 10, pp. 2155–2161, 2010.

[24] A. Campbell and A. S. Wu, "Multi-agent role allocation: Issues, approaches, and multiple perspectives," *Auton. Agents Multi Agent Syst.*, vol. 22, no. 2, pp. 317–355, 2011.

[25] C. Kolias, G. Kambourakis, and M. Maragoudakis, "Swarm intelligence in intrusion detection: A survey," *Comput. Security*, vol. 30, no. 8, pp. 625–642, 2011.

[26] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013.

[27] S. Elsayed, R. Sarker, and D. Essam, "Survey of uses of evolutionary computation algorithms and swarm intelligence for network intrusion detection," *Int. J. Comput. Intell. Appl.*, vol. 14, no. 4, 2015, Art. no. 1550025.

[28] H. Bai and B. Zhao, "A survey on application of swarm intelligence computation to electric power system," in *Proc. 6th World Congr. Intell. Control Autom.*, vol. 2. Dalian, China: IEEE, 2006, pp. 7587–7591.

[29] E. Yang and D. Gu, "Multiagent reinforcement learning for multi-robot systems: A survey," Dept. Comput. Sci., Univ. Essex, Colchester, U.K., Rep. CSM-404, 2004.

[30] N. S. Pal and S. Sharma, "Robot path planning using swarm intelligence: A survey," *Int. J. Comput. Appl.*, vol. 83, no. 12, pp. 5–12, 2013.

[31] A. A. A. Esmin, R. A. Coelho, and S. Matwin, "A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data," *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 23–45, 2015.

[32] D. Martens, B. Baesens, and T. Fawcett, "Editorial survey: Swarm intelligence for data mining," *Mach. Learn.*, vol. 82, no. 1, pp. 1–42, 2011.

[33] M. Bratman, *Intention, Plans, and Practical Reason*. Cambridge, MA, USA: Harvard Univ. Press, 1987.

[34] D. Poole, "Exploiting the rule structure for decision making within the independent choice logic," in *Proc. 11th Conf. Uncertainty Artif. Intell.*, Morgan Kaufmann, 1995, pp. 454–463.

[35] J. Lee, M. J. Huber, E. H. Durfee, and P. G. Kenny, "UM-PRS: An implementation of the procedural reasoning system for multirobot applications," in *Proc. NASA Conf. Publ.*, 1994, pp. 842–849.

[36] D. Poole, "The independent choice logic for modelling multiple agents under uncertainty," *Artif. Intell.*, vol. 94, nos. 1–2, pp. 7–56, 1997.

[37] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Leiden, The Netherlands: Cambridge Univ. Press, 2008.

[38] P. Davidsson, S. Johansson, and M. Svahnberg, "Characterization and evaluation of multi-agent system architectural styles," in *Software Engineering for Multi-Agent Systems IV*. Heidelberg, Germany: Springer, 2005, pp. 179–188.

[39] L. E. Parker, "Multiple mobile robot systems," in *Springer Handbook of Robotics*. Heidelberg, Germany: Springer, 2008, pp. 921–941.

[40] M. Ngobye, W. T. De Groot, and T. P. Van Der Weide, "Types and priorities of multi-agent system interactions," *Interdiscipl. Description Complex Syst.*, vol. 8, no. 1, pp. 49–58, 2010.

[41] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, USA: Pearson, 2009.

[42] P. Langley, J. E. Laird, and S. Rogers, "Cognitive architectures: Research issues and challenges," *Cogn. Syst. Res.*, vol. 10, no. 2, pp. 141–160, 2009.

[43] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, vol. 1. Harlow, U.K.: Addison-Wesley Read., 1999.

[44] L. E. Parker, "Distributed intelligence: Overview of the field and its application in multi-robot systems," *J. Phys. Agents*, vol. 2, no. 1, pp. 5–14, 2008.

[45] E. Altman, *Constrained Markov Decision Processes*, vol. 7. Boca Raton, FL, USA: CRC Press, 1999.

[46] J. A. Boyan and M. L. Littman, "Exact solutions to time-dependent MDPs," *Work*, vol. 2, no. 12, p. 10, 2000.

[47] K. V. Delgado, S. Sanner, and L. N. De Barros, "Efficient solutions to factored MDPs with imprecise transition probabilities," *Artif. Intell.*, vol. 175, nos. 9–10, pp. 1498–1527, 2011.

[48] R. S. Sutton *et al.*, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proc. Int. Conf. Mach. Learn.*, Montreal, QC, Canada: ACM, 2009, pp. 993–1000.

[49] R. S. Sutton, H. R. Maei, and C. Szepesvári, "A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation," in *Proc. NIPS*, 2009, pp. 1609–1616.

[50] P. Escandell-Montero, J. M. Martínez-Martínez, J. D. Martín-Guerrero, E. Soria-Olivas, and J. Gómez-Sanchis, "Least-squares temporal difference learning based on an extreme learning machine," *Neurocomputing*, vol. 141, pp. 37–45, Oct. 2014.

[51] D. S. Bernstein, C. Amato, E. A. Hansen, and S. Zilberstein, "Policy iteration for decentralized control of Markov decision processes," *J. Artif. Intell. Res.*, vol. 34, no. 1, pp. 89–132, 2009.

[52] A. Ghate and R. L. Smith, "A linear programming approach to nonstationary infinite-horizon Markov decision processes," *Oper. Res.*, vol. 61, no. 2, pp. 413–425, 2013.

[53] F. Dufour and T. Prieto-Rumeau, "Finite linear programming approximations of constrained discounted Markov decision processes," *SIAM J. Control Optim.*, vol. 51, no. 2, pp. 1298–1324, 2013.

[54] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Univ. Cambridge, Cambridge, U.K., 1989.

[55] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," vol. 37, Dept. Eng., Univ. Cambridge, Cambridge, U.K., Rep. CUED/F-INFENG/TR-166, 1994.

[56] J. H. Metzen, M. Edgington, Y. Kassahun, and F. Kirchner, "Analysis of an evolutionary reinforcement learning method in a multiagent domain," in *Proc. AAMAS*, vol. 1. Estoril, Portugal, 2008, pp. 291–298.

[57] S. Whiteson, "Evolutionary computation for reinforcement learning," in *Reinforcement Learning*. Heidelberg, Germany: Springer, 2012, pp. 325–355.

[58] J. Schmidhuber, D. Wierstra, and F. Gomez, "Evolino: Hybrid neuroevolution/optimal linear search for sequence learning," in *Proc. 19th Int. Joint Conf. Artif. Intell.*, Morgan Kaufmann, 2005, pp. 853–858.

[59] J. Schmidhuber, D. Wierstra, M. Gagliolo, and F. Gomez, "Training recurrent networks by Evolino," *Neural Comput.*, vol. 19, no. 3, pp. 757–779, Mar. 2007.

[60] J. H. Metzen, F. Kirchner, M. Edgington, and Y. Kassahun, "Towards efficient online reinforcement learning using neuroevolution," in *Proc. 10th Annu. Conf. Genet. Evol. Comput.*, Atlanta, GA, USA, 2008, pp. 1425–1426.

[61] R. Koppejan and S. Whiteson, "Neuroevolutionary reinforcement learning for generalized helicopter control," in *Proc. 11th ACM Annu. Conf. Genet. Evol. Comput.*, Montreal, QC, Canada, 2009, pp. 145–152.

[62] B. Li and J. Si, "Approximate robust policy iteration using multilayer perceptron neural networks for discounted infinite-horizon Markov decision processes with uncertain correlated transition matrices," *IEEE Trans. Neural Netw.*, vol. 21, no. 8, pp. 1270–1280, Aug. 2010.

[63] A. Hans and S. Udluft, "Ensembles of neural networks for robust reinforcement learning," in *Proc. IEEE Int. Conf. Mach. Learn. Appl.*, Washington, DC, USA, 2010, pp. 401–406.

[64] A. Nair *et al.*, "Massively parallel methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn. Deep Learn. Workshop*, Lille, France, 2015, pp. 1–14.

[65] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[66] R. Tutunov, J. El-Zini, H. Bou-Ammar, and A. Jadbabaie, "Distributed lifelong reinforcement learning with sub-linear regret," in *Proc. 56th IEEE Annu. Conf. Decis. Control*, Melbourne, VIC, Australia, 2017, pp. 2254–2259.

[67] J. Fu, S. Han, and U. Topcu, "Optimal control in Markov decision processes via distributed optimization," in *Proc. 54th IEEE Annu. Conf. Decis. Control*, 2015, pp. 7462–7469.

[68] C. Boutilier, "Sequential optimality and coordination in multiagent systems," in *Proc. IJCAI*, vol. 99, 1999, pp. 478–485.

[69] C. Guestrin, D. Koller, and R. Parr, "Multiagent planning with factored MDPS," in *Proc. NIPS*, vol. 1, 2001, pp. 1523–1530.

[70] D. S. Bernstein, S. Zilberstein, and N. Immerman, "The complexity of decentralized control of Markov decision processes," in *Proc. 16th Conf. Uncertainty Artif. Intell.*, Morgan Kaufmann, 2000, pp. 32–37.

[71] S. Brechtel, T. Gindele, and R. Dillmann, "Solving continuous POMDPs: Value iteration with incremental learning of an efficient space representation," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 370–378.

[72] X. P. Wan and S. Y. Li, "SHP-VI method of solving DEC-POMDP problem," *Adv. Mater. Res.*, vol. 926, 2014, pp. 3245–3249.

[73] F. Wu, S. Zilberstein, and N. R. Jennings, "Monte–Carlo expectation maximization for decentralized POMDPs," in *Proc. IJCAI*, 2013, pp. 397–403.

[74] Z. Song, X. Liao, and L. Carin, "Solving DEC-POMDPs by expectation maximization of value function," in *Proc. AAAI Spring Symp. Series*, 2016, pp. 68–76.

[75] C. Amato, D. S. Bernstein, and S. Zilberstein, "Optimizing memory-bounded controllers for decentralized POMDPs," in *Proc. 23rd Conf. Uncertainty Artif. Intell.*, 2007, pp. 1–8.

[76] C. Amato, D. S. Bernstein, and S. Zilberstein, "Solving POMDPs using quadratically constrained linear programs," in *Proc. AAMAS*, ACM, 2006, pp. 341–343.

[77] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo, "Monte–Carlo value iteration for continuous-state POMDPs," in *Algorithmic Foundations of Robotics IX*. Heidelberg, Germany: Springer, 2010, pp. 175–191.

[78] D. Silver and J. Veness, "Monte–Carlo planning in large POMDPs," in *Proc. NIPS*, 2010, pp. 2164–2172.

[79] S. Seuken and S. Zilberstein, "Memory-bounded dynamic programming for DEC-POMDPs," in *Proc. IJCAI*, 2007, pp. 2009–2015.

[80] S. Seuken and S. Zilberstein, "Improved memory-bounded dynamic programming for decentralized POMDPs," in *Proc. 23rd Conf. Uncertainty Artif. Intell.*, 2007, pp. 2009–2015.

[81] D. Szer, F. Charpillet, and S. Zilberstein, "MAA*: A heuristic search algorithm for solving decentralized POMDPs," in *Proc. 21st Conf. Uncertainty Artif. Intell.*, 2005, pp. 576–583.

[82] B. Eker and H. L. Akın, "Solving decentralized POMDP problems using genetic algorithms," *Auton. Agents Multi Agent Syst.*, vol. 27, no. 1, pp. 161–196, 2013.

[83] F. J. Gomez and J. Schmidhuber, "Co-evolving recurrent neurons learn deep memory POMDPs," in *Proc. 7th Annu. Conf. Genet. Evol. Comput.*, ACM, 2005, pp. 491–498.

[84] L. Dung, T. Komeda, and M. Takagi, "Knowledge-based recurrent neural networks in reinforcement learning," in *Proc. 11th Int. Conf. Artif. Intell. Soft Comput.*, ACTA Press, 2007, pp. 169–174.

[85] Z. Liu and I. Elhanany, "A scalable model-free recurrent neural network framework for solving POMDPs," in *Proc. IEEE Int. Symp. Approx. Dyn. Program. Reinforcement Learn.*, 2007, pp. 119–126.

[86] D. Wierstra, A. Foerster, J. Peters, and J. Schmidhuber, "Solving deep memory POMDPs with recurrent policy gradients," in *Artificial Neural Networks*. Heidelberg, Germany: Springer, 2007, pp. 697–706.

[87] M. Liu, X. Liao, and L. Carin, "The infinite regionalized policy representation," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 769–776.

[88] G. Contardo, L. Denoyer, T. Artieres, and P. Gallinari, "Learning states representations in POMDP," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 120–122.

[89] M. Lauri and R. Ritala, "Planning for robotic exploration based on forward simulation," *Robot. Auton. Syst.*, vol. 83, pp. 15–31, Sep. 2016.

[90] D. S. Bernstein, E. A. Hansen, and S. Zilberstein, "Bounded policy iteration for decentralized POMDPs," in *Proc. 19th Int. Joint Conf. Artif. Intell.*, 2005, pp. 1287–1292.

[91] F. A. Oliehoek, M. T. J. Spaan, and N. A. Vlassis, "Optimal and approximate Q-value functions for decentralized POMDPs," *J. Artif. Intell. Res.*, vol. 32, no. 1, pp. 289–353, 2008.

[92] D. Szer and F. Charpillet, "Point-based dynamic programming for Dec-POMDPs," in *Proc. AAAI*, vol. 6, 2006, pp. 1233–1238.

[93] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *Proc. AAAI*, vol. 4, 2004, pp. 709–715.

[94] A. Kumar and S. Zilberstein, "Dynamic programming approximations for partially observable stochastic games," in *Proc. 22nd Int. FLAIRS Conf.*, 2009, pp. 547–552.

[95] M. Liu, C. Amato, X. Liao, L. Carin, and J. P. How, "Stick-breaking policy learning in Dec-POMDPs," in *Proc. 24th Int. Conf. Artif. Intell.*, AAAI Press, 2015, pp. 2011–2017.

[96] F. A. Oliehoek, S. Whiteson, and M. T. J. Spaan, "Approximate solutions for factored Dec-POMDPs with many agents," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2013, pp. 563–570.

[97] D. V. Pynadath and M. Tambe, "The communicative multiagent team decision problem: Analyzing teamwork theories and models," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 389–423, 2002.

[98] S. Seuken and S. Zilberstein, "Formal models and algorithms for decentralized decision making under uncertainty," *Auton. Agents Multi Agent Syst.*, vol. 17, no. 2, pp. 190–250, 2008.

[99] C. Amato, F. A. Oliehoek, and E. Shyu, "Scalable Bayesian reinforcement learning for multiagent POMDPs," in *Proc. 1st Multidiscipl. Conf. Reinforcement Learn. Decis. Making*, 2013, pp. 188–192.

[100] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo, "Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs," in *Proc. AAAI*, vol. 5, 2005, pp. 133–139.

[101] C. Zhang and V. R. Lesser, "Coordinated multi-agent reinforcement learning in networked distributed POMDPs," in *Proc. AAAI*, 2011, pp. 764–770.

[102] A. Kumar and S. Zilberstein, "Constraint-based dynamic programming for decentralized POMDPs with structured interactions," in *Proc. AAMAS*, vol. 1, 2009, pp. 561–568.

[103] P. J. Gmytrasiewicz and P. Doshi, "Interactive POMDPs: Properties and preliminary results," in *Proc. AAMAS* , vol. 3, 2004, pp. 1374–1375.

[104] P. Doshi and P. J. Gmytrasiewicz, "A particle filtering based approach to approximating interactive POMDPs," in *Proc. AAAI*, 2005, pp. 969–974.

[105] P. Doshi and D. Perez, "Generalized point based value iteration for interactive POMDPs," in *Proc. AAAI*, 2008, pp. 63–68.

[106] P. Doshi and P. J. Gmytrasiewicz, "Monte–Carlo sampling methods for approximating interactive POMDPs," *J. Artif. Intell. Res.*, pp. 297–337, 2009.

[107] R. B. Myerson, *Game Theory: Analysis of Conflict*. Cambridge, MA, USA: Harvard Univ., 1991.

[108] L. S. Shapley, "Stochastic games," *Proc. Nat. Acad. Sci. USA*, vol. 39, no. 10, pp. 1095–1100, 1953.

[109] A. Condon, "The complexity of stochastic games," *Inf. Comput.*, vol. 96, no. 2, pp. 203–224, 1992.

[110] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *J. Mach. Learn. Res.*, vol. 4, pp. 1039–1069, Dec. 2003.

[111] J. F. Nash, "Equilibrium points in n-person games," *Proc. Nat. Acad. Sci. USA*, vol. 36, no. 1, pp. 48–49, 1950.

[112] J. M. Smith and G. R. Price, "The logic of animal conflict," *Nature*, vol. 246, no. 5427, pp. 15–18, 1973.
[113] J. Flesch, T. Parthasarathy, F. Thuijsman, and P. Uyttendaele, "Evolutionary stochastic games," *Dyn. Games Appl.*, vol. 3, no. 2, pp. 207–219, 2013.
[114] S. Loertscher, "Rock–scissors–paper and evolutionarily stable strategies," *Econ. Lett.*, vol. 118, no. 3, pp. 473–474, 2013.
[115] R. Emery, *Game Theoretic Control for Robot Teams*, document CMU-RI-TR-05-36, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2005.
[116] A. Guo and V. R. Lesser, "Planning for weakly-coupled partially observable stochastic games," in *Proc. IJCAI*, 2005, pp. 1715–1716.
[117] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun, "Approximate solutions for partially observable stochastic games with common payoffs," in *Proc. AAMAS*, 2004, pp. 136–143.
[118] P. Paruchuri *et al.*, "Playing games for security: An efficient exact algorithm for solving Bayesian stackelberg games," in *Proc. AAMAS*, vol. 2, 2008, pp. 895–902.
[119] C. Kiekintveld, J. Marecki, and M. Tambe, "Approximation methods for infinite Bayesian–Stackelberg games: Modeling distributional payoff uncertainty," in *Proc. AAMAS*, vol. 3, 2011, pp. 1005–1012.
[120] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots & Biological Systems: Towards a New Bionics?* Heidelberg, Germany: Springer, 1993, pp. 703–712.
[121] G. Weiss, Ed., *Multiagent Systems*. Cambridge, MA, USA: MIT Press, 2013.
[122] D. Teodorović and M. Dell'orco, "Bee colony optimization—A cooperative learning approach to complex transportation problems," in *Proc. 16th Mini Eur. Conf. 10th Meeting Adv. OR AI Methods Transport. (EWGT)*, 2005, pp. 51–60.
[123] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY, USA: Oxford Univ. Press, 1999.
[124] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.
[125] H. Duan and P. Qiao, "Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning," *Int. J. Intell. Comput. Cybern.*, vol. 7, no. 1, pp. 24–37, 2014.
[126] S. Ilie and C. Bădică, "Multi-agent approach to distributed ant colony optimization," *Sci. Comput. Program.*, vol. 78, no. 6, pp. 762–774, 2013.
[127] J. Collings and E. Kim, "A distributed and decentralized approach for ant colony optimization with fuzzy parameter adaptation in traveling salesman problem," in *Proc. IEEE Symp. Swarm Intell.*, 2014, pp. 1–9.
[128] I. Navarro, E. Di Mario, and A. Martinoli, "Distributed vs. centralized particle swarm optimization for learning flocking behaviors," in *Proc. Eur. Conf. Synth. Simulat. Living Syst.*, 2015, pp. 302–309.
[129] M. Dorigo *et al.*, "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 60–71, Dec. 2013.
[130] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
[131] R. A. Howard and J. E. Matheson, "Influence diagrams," *Decis. Anal.*, vol. 2, no. 3, pp. 127–143, 2005.
[132] J. A. Tatman and R. D. Shachter, "Dynamic programming and influence diagrams," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 365–379, Mar./Apr. 1990.
[133] Y. Zeng, P. Doshi, and Q. Chen, "Approximate solutions of interactive dynamic influence diagrams using model clustering," in *Proc. Nat. Conf. Artif. Intell.*, vol. 22. Vancouver, BC, Canada, 2007, pp. 782–787.
[134] D. Koller and B. Milch, "Multi-agent influence diagrams for representing and solving games," *Games Econ. Behav.*, vol. 45, no. 1, pp. 181–221, 2003.
[135] Y. Gal and A. Pfeffer, "Networks of influence diagrams: A formalism for representing agents' beliefs and decision-making processes," *J. Artif. Intell. Res.*, vol. 33, no. 1, pp. 109–147, 2008.
[136] D. K. Molzahn *et al.*, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Nov. 2017.
[137] A. Bidram, A. Davoudi, F. L. Lewis, and Z. Qu, "Secondary control of microgrids based on distributed cooperative control of multi-agent systems," *IET Gener. Transm. Distrib.*, vol. 7, no. 8, pp. 822–831, Aug. 2013.
[138] J. Ni, L. Liu, C. Liu, X. Hu, and S. Li, "Secondary voltage control for microgrids based on fixed-time distributed cooperative control of multi-agent systems," in *Proc. IEEE Amer. Control Conf.*, Seattle, WA, USA, 2017, pp. 761–766.
[139] Y.-J. Pan, H. Werner, Z. Huang, and M. Bartels, "Distributed cooperative control of leader–follower multi-agent systems under packet dropouts for quadcopters," *Syst. Control Lett.*, vol. 106, pp. 47–57, Aug. 2017.
[140] W. Wang, J. Huang, and C. Wen, "Distributed adaptive control of multi-agent systems under directed graph for asymptotically consensus tracking," in *Proc. 14th Int. Conf. Control Autom. Robot. Vis.*, 2016, pp. 1–5.
[141] H. Zhang, T. Feng, G.-H. Yang, and H. Liang, "Distributed cooperative optimal control for multiagent systems on directed graphs: An inverse optimal approach," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1315–1326, Jul. 2015.
[142] W. He, G. Chen, Q.-L. Han, and F. Qian, "Network-based leader-following consensus of nonlinear multi-agent systems via distributed impulsive control," *Inf. Sci.*, vol. 380, pp. 145–158, Feb. 2017.
[143] W. He *et al.*, "Leader-following consensus of nonlinear multiagent systems with stochastic sampling," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 327–338, Feb. 2017.
[144] W. He *et al.*, "Quasi-synchronization of heterogeneous dynamic networks via distributed impulsive control: Error estimation, optimization and design," *Automatica*, vol. 62, pp. 249–262, Dec. 2015.
[145] R. Yang, H. Zhang, H. Yan, and F. Yang, "Cooperative control of heterogeneous multi-agent systems via distributed adaptive output regulation under switching topology," in *Proc. 42nd Annu. IEEE Conf. Ind. Electron. Soc.*, Florence, Italy, 2016, pp. 6770–6775.
[146] G. Wang, C. Wang, Q. Du, L. Li, and W. Dong, "Distributed cooperative control of multiple nonholonomic mobile robots," *J. Intell. Robot. Syst.*, vol. 83, nos. 3–4, pp. 525–541, 2016.
[147] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein, "Graph-based distributed cooperative navigation for a general multi-robot measurement model," *Int. J. Robot. Res.*, vol. 31, no. 9, pp. 1057–1080, 2012.
[148] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun, "Game theoretic control for robot teams," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 1163–1169.
[149] C. Murray and G. J. Gordon, "Multi-robot negotiation: Approximating the set of subgame perfect equilibria in general-sum stochastic games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1001–1008.
[150] Y.-C. Choi and H.-S. Ahn, "A survey on multi-agent reinforcement learning: Coordination problems," in *Proc. IEEE/ASME Int. Conf. Mechatronics Embedded Syst. Appl.*, Qingdao, China, 2010, pp. 81–86.
[151] S. Ragi and E. K. P. Chong, "UAV path planning in a dynamic environment via partially observable Markov decision process," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2397–2412, Oct. 2013.
[152] C. Amato *et al.*, "Probabilistic planning for decentralized multi-robot systems," in *Proc. AAAI Fall Symp. Series*, 2015, pp. 10–12.
[153] C. Amato *et al.*, "Planning for decentralized control of multiple robots under uncertainty," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seattle, WA, USA, 2015, pp. 1241–1248.
[154] C. Amato *et al.*, "Policy search for multi-robot coordination under uncertainty," in *Proc. Robot. Sci. Syst. Conf.*, 2015, pp. 1760–1778.
[155] D. K. Grady, M. Moll, and L. E. Kavraki, "Extending the applicability of POMDP solutions to robotic tasks," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 948–961, Aug. 2015.
[156] O. Aşık and H. L. Akın, "Solving multi-agent decision problems modeled as Dec-POMDP: A robot soccer case study," in *RoboCup 2012: Robot Soccer World Cup XVI*. Heidelberg, Germany: Springer, 2013, pp. 130–140.
[157] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
[158] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, Aug. 2007.
[159] F. Xiao, L. Wang, J. Chen, and Y. Gao, "Finite-time formation control for multi-agent systems," *Automatica*, vol. 45, no. 11, pp. 2605–2611, 2009.
[160] M. A. Joordens and M. Jamshidi, "Consensus control for a system of underwater swarm robots," *IEEE Syst. J.*, vol. 4, no. 1, pp. 65–73, Mar. 2010.
[161] R. M. de Mendonça, N. Nedjah, and L. de Macedo Mourelle, "Efficient distributed algorithm of dynamic task assignment for swarm robotics," *Neurocomputing*, vol. 172, pp. 345–355, Jan. 2016.
[162] A. O. de Sá, N. Nedjah, and L. de Macedo Mourelle, "Distributed efficient localization in swarm robotic systems using swarm intelligence algorithms," *Neurocomputing*, vol. 172, pp. 322–336, Jan. 2016.

[163] S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Teamwork in self-organized robot colonies," *IEEE Trans. Evol. Comput*, vol. 13, no. 4, pp. 695–711, Aug. 2009.
[164] S. Alers, D. Claes, K. Tuyls, and G. Weiss, "Biologically inspired multi-robot foraging," in *Proc. AAMAS*, Paris, France, 2014, pp. 1683–1684.
[165] K. Ohkura, T. Yasuda, Y. Matsumura, and M. Kadota, "GPU implementation of food-foraging problem for evolutionary swarm robotics systems," in *Swarm Intelligence*. Cham, Switzerland: Springer, 2014, pp. 238–245.
[166] T. Schmickl *et al.*, "Get in touch: Cooperative decision making based on robot-to-robot collisions," *Auton. Agents Multi Agent Syst.*, vol. 18, no. 1, pp. 133–155, 2009.
[167] H. Wei, Y. Cai, H. Li, D. Li, and T. Wang, "Sambot: A self-assembly modular robot for swarm robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, 2010, pp. 66–71.
[168] C. Moeslinger, T. Schmickl, and K. Crailsheim, "A minimalist flocking algorithm for swarm robots," in *Advances in Artificial Life. Darwin Meets von Neumann*. Heidelberg, Germany: Springer, 2011, pp. 375–382.
[169] F. Mondada *et al.*, "The cooperation of swarm-bots: Physical interactions in collective robotics," *IEEE Robot. Autom. Mag.*, vol. 12, no. 2, pp. 21–28, Jun. 2005.
[170] R. O'Grady *et al.*, "Swarm-bots to the rescue," in *Advances in Artificial Life. Darwin Meets von Neumann*. Heidelberg, Germany: Springer, 2011, pp. 165–172.
[171] S. Kernbach *et al.*, "Adaptive collective decision-making in limited robot swarms without communication," *Int. J. Robot. Res.*, vol. 32, no. 1, pp. 35–55, 2013.
[172] T. Matthews, S. D. Ramchurn, and G. Chalkiadakis, "Competing with humans at fantasy football: Team formation in large partially-observable domains," in *Proc. AAAI*, Toronto, ON, Canada, 2012, pp. 1394–1400.
[173] S. S. Liao, J.-D. Zhang, R. Lau, and T. Wu, "Coalition formation based on marginal contributions and the Markov process," *Decis. Support Syst.*, vol. 57, pp. 355–363, Jan. 2014.
[174] G. Chalkiadakis and C. Boutilier, "Sequentially optimal repeated coalition formation under uncertainty," *Auton. Agents Multi Agent Syst.*, vol. 24, no. 3, pp. 441–484, 2012.
[175] S. D. Sen and J. A. Adams, "An influence diagram based multi-criteria decision making framework for multirobot coalition formation," *Auton. Agents Multi Agent Syst.*, vol. 29, no. 6, pp. 1061–1090, 2015.
[176] D. S. D. Santos and A. L. C. Bazzan, "Distributed clustering for group formation and task allocation in multiagent systems: A swarm intelligence approach," *Appl. Soft Comput.*, vol. 12, no. 8, pp. 2123–2131, 2012.
[177] S. Ieong and Y. Shoham, "Bayesian coalitional games," in *Proc. AAAI*, Chicago, IL, USA, 2008, pp. 95–100.
[178] K. Akkarajitsakul, E. Hossain, and D. Niyato, "Coalition-based cooperative packet delivery under uncertainty: A dynamic Bayesian coalitional game," *IEEE Trans. Mobile Comput.*, vol. 12, no. 2, pp. 371–385, Feb. 2013.
[179] Y. Zeng, K. Xiang, D. Li, and A. V. Vasilakos, "Directional routing and scheduling for green vehicular delay tolerant networks," *Wireless Netw.*, vol. 19, no. 2, pp. 161–173, 2013.
[180] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," in *Proc. 16th Int. IEEE Conf. Intell. Transport. Syst.*, The Hague, The Netherlands, 2013, pp. 2063–2067.
[181] S. A. Kulkarni and G. R. Rao, "Vehicular ad hoc network mobility models applied for reinforcement learning routing algorithm," in *Contemporary Computing*. Heidelberg, Germany: Springer, 2010, pp. 230–240.
[182] C. Wu, S. Ohzahata, and T. Kato, "Flexible, portable, and practicable solution for routing in VANETs: A fuzzy constraint Q-learning approach," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4251–4263, Nov. 2013.
[183] M. Boushaba, A. Hafid, A. Belbekkouche, and M. Gendreau, "Reinforcement learning based routing in wireless mesh networks," *Wireless Netw.*, vol. 19, no. 8, pp. 2079–2091, 2013.
[184] S. Hosseininezhad, G. N. Shirazi, and V. C. M. Leung, "RLAB: Reinforcement learning-based adaptive broadcasting for vehicular ad-hoc networks," in *Proc. IEEE 73rd Veh. Technol. Conf.*, Yokohama, Japan, 2011, pp. 1–5.
[185] A. Soua and H. Afifi, "Adaptive data collection protocol using reinforcement learning for VANETs," in *Proc. 9th Int. Wireless Commun. Mobile Comput. Conf.*, 2013, pp. 1040–1045.
[186] A. L. C. Bazzan, "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control," *Auton. Agents Multi Agent Syst.*, vol. 18, no. 3, pp. 342–375, 2009.
[187] D. Houli, L. Zhiheng, and Z. Yi, "Multiobjective reinforcement learning for traffic signal control using vehicular ad hoc network," *EURASIP J. Adv. Signal Process.*, vol. 1, p. 7, Dec. 2010.
[188] P. G. Balaji, X. German, and D. Srinivasan, "Urban traffic signal control using reinforcement learning agents," *IET Intell. Transp. Syst.*, vol. 4, no. 3, pp. 177–188, Sep. 2010.
[189] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Traffic light control in non-stationary environments based on multi agent Q-learning," in *Proc. 14th Int. IEEE Conf. Intell. Transport. Syst.*, Washington, DC, USA, 2011, pp. 1580–1585.
[190] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 128–135, Jun. 2010.
[191] S. El-Tantawy and B. Abdulhai, "Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC)," in *Proc. Int. Conf. Intell. Transport. Syst.*, Anchorage, AK, USA, 2012, pp. 319–326.
[192] M. A. Khamis and W. Gomaa, "Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework," *Eng. Appl. Artif. Intell.*, vol. 29, pp. 134–151, Mar. 2014.
[193] K. J. Prabuchandran, A. N. H. Kumar, and S. Bhatnagar, "Multi-agent reinforcement learning for traffic signal control," in *Proc. IEEE 17th Int. Conf. Intell. Transport. Syst.*, Qingdao, China, 2014, pp. 2529–2534.
[194] J. García-Nieto, E. Alba, and A. C. Olivera, "Swarm intelligence for traffic light scheduling: Application to real urban areas," *Eng. Appl. Artif. Intell.*, vol. 25, no. 2, pp. 274–283, 2012.
[195] D. Teodorović, "Transport modeling by multi-agent systems: A swarm intelligence approach," *Transport. Plan. Technol.*, vol. 26, no. 4, pp. 289–312, 2003.
[196] B. Tatomir and L. Rothkrantz, "Hierarchical routing in traffic using swarm-intelligence," in *Proc. IEEE Intell. Transport. Syst. Conf.*, Toronto, ON, Canada, 2006, pp. 230–235.
[197] D. Teodorović, "Swarm intelligence systems for transportation engineering: Principles and applications," *Transport. Res. C Emerg. Technol.*, vol. 16, no. 6, pp. 651–667, 2008.
[198] R. Muraleedharan and L. A. Osadciw, "Cognitive security protocol for sensor based VANET using swarm intelligence," in *Proc. 43rd Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, USA: IEEE, 2009, pp. 288–290.
[199] S. S. Manvi, M. S. Kakkasageri, and C. V. Mahapurush, "Performance analysis of AODV, DSR, and swarm intelligence routing protocols in vehicular ad hoc network environment," in *Proc. IEEE Int. Conf. Future Comput. Commun.*, Kuala Lumpur, Malaysia, 2009, pp. 21–25.
[200] J. Toutouh and E. Alba, "Parallel swarm intelligence for VANETs optimization," in *Proc. 7th IEEE Int. Conf. P2P Parallel Grid Cloud Internet Comput.*, Victoria, BC, Canada, 2012, pp. 285–290.
[201] P. Wararkar and S. S. Dorle, "Vehicular adhoc networks handovers with metaheuristic algorithms," in *Proc. IEEE Int. Conf. Electron. Syst. Signal Process. Comput. Technol.*, 2014, pp. 160–165.
[202] S. Medetov, M. Bakhouya, J. Gaber, K. Zinedine, and M. Wack, "A bee-inspired approach for information dissemination in VANETs," in *Proc. IEEE Int. Conf. Multimedia Comput. Syst.*, Marrakesh, Morocco, 2014, pp. 849–854.
[203] M. A. Alsheikh, D. T. Hoang, D. Niyato, H.-P. Tan, and S. Lin, "Markov decision processes with applications in wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1239–1267, 3rd Quart., 2015.
[204] D. L. Kovacs, W. Li, N. Fukuta, and T. Watanabe, "Mixed observability Markov decision processes for overall network performance optimization in wireless sensor networks," in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Fukuoka, Japan, 2012, pp. 289–298.
[205] J. Pajarinen, A. Hottinen, and J. Peltonen, "Optimizing spatial and temporal reuse in wireless networks by decentralized partially observable Markov decision processes," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 866–879, Apr. 2014.
[206] J. Zhang, L. Xu, S. Zhou, and X. Ye, "A novel sleep scheduling scheme in green wireless sensor networks," *J. Supercomput.*, vol. 71, no. 3, pp. 1067–1094, 2015.
[207] J. S. Sandhu, A. M. Agogino, and A. K. Agogino "Wireless sensor networks for commercial lighting control: Decision making with multi-agent systems," in *Proc. AAAI Workshop Sensor Netw.*, vol. 10, 2004, pp. 131–140.

[208] S. S. Iyengar, H.-C. Wu, N. Balakrishnan, and S. Y. Chang, "Biologically inspired cooperative routing for wireless mobile sensor networks," *IEEE Syst. J.*, vol. 1, no. 1, pp. 29–37, Sep. 2007.

[209] M. Farooq, "A comprehensive survey of nature-inspired routing protocols," in *Bee-Inspired Protocol Engineering*. Heidelberg, Germany: Springer, 2009, pp. 19–52.

[210] F. Çelik, A. Zengin, and S. Tuncel, "A survey on swarm intelligence based routing protocols in wireless sensor networks," *Int. J. Phys. Sci.*, vol. 5, no. 14, pp. 2118–2126, 2010.

[211] F. Ducatelle, G. A. Di Caro, and L. M. Gambardella, "Principles and applications of swarm intelligence for adaptive routing in telecommunications networks," *Swarm Intell.*, vol. 4, no. 3, pp. 173–198, 2010.

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

[213] 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," *J. Netw. Comput. Appl.*, vol. 35, no. 5, pp. 1508–1536, 2012.

[214] C. Öztürk, D. Karaboğa, and B. Görkemli, "Artificial bee colony algorithm for dynamic deployment of wireless sensor networks," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 20, no. 2, pp. 255–262, 2012.

[215] H. Ren and M. Q.-H. Meng, "Biologically inspired approaches for wireless sensor networks," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, Luoyang, China, 2006, pp. 762–768.

[216] B. Singh and D. K. Lobiyal, "A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks," *Human Centric Comput. Inf. Sci.*, vol. 2, no. 1, pp. 1–18, 2012.

[217] P. K. Krishnappa and B. P. Babu, "Investigating open issues in swarm intelligence for mitigating security threats in manet," *Int. J. Elect. Comput. Eng.*, vol. 5, no. 5, pp. 1194–1201, 2015.

[218] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 2, pp. 262–267, Mar. 2011.

[219] W.-H. Liao, Y. Kao, and Y.-S. Li, "A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks," *Exp. Syst. Appl.*, vol. 38, no. 10, pp. 12180–12188, 2011.

[220] S. Roy *et al.*, "A survey of game theory as applied to network security," in *Proc. 43rd Hawaii Int. Conf. Syst. Sci.*, Honolulu, HI, USA: IEEE, 2010, pp. 1–10.

[221] A. A. Strikos, *A Full Approach for Intrusion Detection in Wireless Sensor Networks*, School Inf. Commun. Technol., Stockholm, Sweden, 2007.

[222] K. Premkumar and A. Kumar, "Optimal sleep-wake scheduling for quickest intrusion detection using wireless sensor networks," in *Proc. IEEE 27th Conf. Comput. Commun.*, Phoenix, AZ, USA, 2008, pp. 1400–1408.

[223] Y. Liu, C. Comaniciu, and H. Man, "A Bayesian game approach for intrusion detection in wireless ad hoc networks," in *Proc. Workshop Game Theory Commun. Netw.*, Pisa, Italy: ACM, 2006, p. 4.

[224] A. Servin and D. Kudenko, "Multi-agent reinforcement learning for intrusion detection: A case study and evaluation," in *Multiagent System Technologies*. Heidelberg, Germany: Springer, 2008, pp. 159–170.

[225] F. Kuang, S. Zhang, Z. Jin, and W. Xu, "A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection," *Soft Comput.*, vol. 19, no. 5, pp. 1187–1199, 2015.

[226] L. Wang, C. Dong, J. Hu, and G. Li, "Network intrusion detection using support vector machine based on particle swarm optimization," in *Proc. Int. Conf. Appl. Sci. Eng. Innov.*, 2015, pp. 665–670.

[227] A. A. Aburomman and M. B. I. Reaz, "A novel SVM-KNN-PSO ensemble method for intrusion detection system," *Appl. Soft Comput.*, vol. 38, pp. 360–372, Jan. 2016.

[228] S. M. HosseiniBamakan, B. Amiri, M. Mirzabagheri, and Y. Shi, "A new intrusion detection approach using PSO based multiple criteria linear programming," *Procedia Comput. Sci.*, vol. 55, pp. 231–237, Jan. 2015.

[229] P. Wang, H.-T. Lin, and T.-S. Wang, "An improved ant colony system algorithm for solving the IP traceback problem," *Inf. Sci.*, vol. 326, pp. 172–187, Jan. 2016.

[230] C. Wu, K. Chowdhury, M. Di Felice, and W. Meleis, "Spectrum management of cognitive radio using multi-agent reinforcement learning," in *Proc. AAMAS Ind. Track*, Toronto, ON, Canada, 2010, pp. 1705–1712.

[231] H. Li, "Multiagent Q-learning for aloha-like spectrum access in cognitive radio systems," *EURASIP J. Wireless Commun. Netw.*, vol. 1, p. 56, Apr. 2010.

[232] K.-L. A. Yau, P. Komisarczuk, and P. D. Teal, "Applications of reinforcement learning to cognitive radio networks," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2010, pp. 1–6.

[233] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1823–1834, May 2010.

[234] J. Lundén, S. R. Kulkarni, V. Koivunen, and H. V. Poor, "Multiagent reinforcement learning based spectrum sensing policies for cognitive radio networks," *EEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 858–868, Oct. 2013.

[235] Q. Zhu, H. Li, Z. Han, and T. Basar, "A stochastic game model for jamming in multi-channel cognitive radio systems," in *Proc. IEEE Int. Conf. Commun.*, 2010, pp. 1–6.

[236] B. Wang, Y. Wu, and K. J. R. Liu, "Game theory for cognitive radio networks: An overview," *Comput. Netw.*, vol. 54, no. 14, pp. 2537–2561, 2010.

[237] Q. Zhu, Z. Han, and T. Başar, "A differential game approach to distributed demand side management in smart grid," in *Proc. IEEE Int. Conf. Commun.*, Ottawa, ON, Canada, 2012, pp. 3345–3350.

[238] F. S. Melo, A. Sardinha, S. Witwicki, L. M. Ramirez-Elizondo, and M. T. Spaan, "Decentralized multiagent planning for balance control in smart grids," in *Proc. 1st Int. Workshop Inf. Technol. Energy Appl.*, vol. 923, 2012, pp. 21–24.

[239] J. Wu and X. Guan, "Coordinated multi-microgrids optimal control algorithm for smart distribution management system," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2174–2181, Dec. 2013.

[240] F. Lauri *et al.*, "Managing power flows in microgrids using multi-agent reinforcement learning," in *Proc. Agent Technol. Energy Syst.*, 2013, pp. 1–8.

[241] M. Peters, W. Ketter, M. Saar-Tsechansky, and J. Collins, "A reinforcement learning approach to autonomous decision-making in smart electricity markets," *Mach. Learn.*, vol. 92, no. 1, pp. 5–39, 2013.

[242] Y. Huang, A. Brocco, P. Kuonen, M. Courant, and B. Hirsbrunner, "SmartGRID: A fully decentralized grid scheduling framework supported by swarm intelligence," in *Proc. 7th IEEE Int. Conf. Grid Cooperative Comput.*, Shenzhen, China, 2008, pp. 160–168.

[243] P. Faria, Z. A. Vale, J. Soares, and J. Ferreira, "Particle swarm optimization applied to integrated demand response resources scheduling," in *Proc. IEEE Symp. Comput. Intell. Appl. Smart Grid*, Paris, France, 2011, pp. 1–8.

[244] Q. Kang, M. Zhou, J. An, and Q. Wu, "Swarm intelligence approaches to optimal power flow problem with distributed generator failures in power networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 2, pp. 343–353, Apr. 2013.

[245] M. Hemmati, A. Yassine, and S. Shirmohammadi, "A Dec-POMDP model for congestion avoidance and fair allocation of network bandwidth in rate-adaptive video streaming," in *Proc. IEEE Symp. Series Comput. Intell.*, 2015, pp. 1182–1189.

[246] F. Teng and F. Magoulès, "A new game theoretical resource allocation algorithm for cloud computing," in *Advances in Grid and Pervasive Computing*. Heidelberg, Germany: Springer, 2010, pp. 321–330.

[247] J. Wu, X. Xu, P. Zhang, and C. Liu, "A novel multi-agent reinforcement learning approach for job scheduling in grid computing," *Future Gener. Comput. Syst.*, vol. 27, no. 5, pp. 430–439, 2011.

[248] S. Yang, Q. Liu, and J. Wang, "Distributed optimization based on a multiagent system in the presence of communication delays," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 5, pp. 717–728, May 2017.

[249] W. Stadler, "Fundamentals of multicriteria optimization," in *Multicriteria Optimization in Engineering and in the Sciences*. Boston, MA, USA: Springer, 1988, pp. 1–25.

[250] T. L. Vincent, "Game theory as a design tool," *J. Mech. Transm. Autom. Design*, vol. 105, no. 2, pp. 165–170, 1983.

[251] J. Zhang, D. Qi, and G. Zhao, "A game theoretical formulation for distributed optimization problems," in *Proc. 10th IEEE Int. Conf. Control Autom.*, Hangzhou, China, 2013, pp. 1939–1944.

[252] K. Hasegawa and M. Noto, "Swarm intelligence algorithm for optimality discovery in distributed constraint optimization," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, San Diego, CA, USA, 2014, pp. 3611–3616.

[253] B. Cao *et al.*, "Distributed parallel particle swarm optimization for multi-objective and many-objective large-scale optimization," *IEEE Access*, vol. 5, pp. 8214–8221, 2017.

[254] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst.*, vol. 22, no. 3, pp. 52–67, Jun. 2002.
[255] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.
[256] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 781–786, Mar. 2014.
[257] L. Braubach, A. Pokahr, and W. Lamersdorf, "A universal criteria catalog for evaluation of heterogeneous agent development artifacts," in *Proc. From Agent Theory Agent Implement.*, 2008, pp. 19–28.
[258] R. N. Lass, E. A. Sultanik, and W. C. Regli, "Metrics for multiagent systems," in *Performance Evaluation and Benchmarking of Intelligent Systems*. Boston, MA, USA: Springer, 2009, pp. 1–19.
[259] P. Di Bitonto, M. Laterza, T. Roselli, and V. Rossano, "An evaluation method for multi-agent systems," in *Agent and Multi-Agent Systems: Technologies and Applications*. Heidelberg, Germany: Springer, 2010, pp. 32–41.
[260] T. Marir, F. Mokhati, H. Bouchelaghem-Seridi, and B. Benaissa, "Dynamic metrics for multi-agent systems using aspect-oriented programming," in *Proc. German Conf. Multiagent Syst. Technol.*, Klagenfurt, Austria: Springer, 2016, pp. 58–72.

Yara Rizk (M'09) received the B.E. degree in computer and communication engineering from the American University of Beirut, Beirut, Lebanon, in 2012, where she is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department.

She was a Technical Intern with Intel, Hillsboro, OR, USA, from 2013 to 2014. She is an active researcher with multiple peer-reviewed publications. Her current research interests include robotics, multiagent systems, machine learning, classification, clustering, and artificial intelligence.

Mariette Awad (M'08) received the Ph.D. degree in electrical engineering from the University of Vermont, Burlington, VT, USA, in 2007.

She was a Wireless Product Engineer with IBM System and Technology Group, Essex, VT, USA, for six years, where her technical leadership and innovative spirit has earned her management recognition twice, two business awards, and ten patents. She has authored a book entitled *Efficient Machine Learning* (2015) as well as over 70 conference, book chapter, and journal publications. Her current research interests include human machine interface, artificial intelligence, machine learning, data analytics, and Internet of Things.

Dr. Awad was a recipient of 21 grants to support her research including two multidisciplinary multimillion dollar grants from Qatar National Research Fund and Intel. She has been an invited speaker and a reviewer at a number of international conferences and journals.

Edward W. Tunstel (S'92–M'96–SM'99–F'12) received the B.S. and M.Eng. degrees in mechanical engineering from Howard University, Washington, DC, USA, in 1986 and 1989, respectively, and the Ph.D. degree in electrical engineering from the University of New Mexico, Albuquerque, NM, USA, in 1996.

He was a Senior Robotics Engineer with NASA Jet Propulsion Laboratory, Pasadena, CA, USA, from 1989 to 2007, and a Space Robotics and Autonomous Control Lead and a Senior Roboticist with Johns Hopkins Applied Physics Laboratory, Laurel, MD, USA, from 2007 to 2017. He is currently an Associate Director of Robotics with United Technologies Research Center, East Hartford, CT, USA. His current research interests include mobile robot navigation, autonomous control, cooperative multirobot systems, human-collaborative robotics, robotic systems engineering, and applications of soft computing to autonomous systems.

Dr. Tunstel has been serving as the President of the IEEE Systems, Man, and Cybernetics Society since 2018.