IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE AND COGNITIVE COMPUTING FOR COMMUNICATION AND NETWORK

Received June 13, 2019, accepted June 21, 2019, date of publication July 4, 2019, date of current version August 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926286

iAgent: When AI Meets Mobile Agent

JIAYI LU[1], WENJING XIAO[1], ENMIN SONG[1], MOHAMMAD MEHEDI HASSAN[2], AHMAD ALMOGREN[2], AND AYMAN ALTAMEEM[3]

[1]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China 430074
[2]College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[3]Department of Natural and Engineering Sciences, College of Applied Studies and Community Services, King Saud University, Riyadh 11543, Saudi Arabia

There is't any funding or financial support. Corresponding author: Enmin Song (esong@hust.edu.cn)

This work was supported by the Deanship of Scientific Research at King Saud University under Grant RGP-1437-35.

ABSTRACT The past few decades have seen the rapid development of artificial intelligence (AI) technology in diverse industries. Inspired by the fact that an agent is the central part of AI, we combine AI with the technology of the mobile agent. A mobile agent used in wireless sensor networks (WSNs) is popular for its mobility, executability, and autonomy. To improve the intelligence of the mobile agent, we proposed a conceptual theoretical framework named iAgent, where i means intelligent, and the agent refers to the mobile agent. Four designs of iAgent are detailed. Compared with the old mobile agent, the iAgent has a learning ability, which means that it can dynamically plan the path according to the external environment in order to reduce energy consumption. Based on iAgent, we also proposed a method to determine the number of iAgents and their visiting areas in a multi-iAgent WSN environment. The extensive simulation indicates that the multi-iAgent algorithm can significantly improve the performance of the WSNs, especially in saving energy and balancing network load.

INDEX TERMS Mobile agent, artificial intelligence, wireless sensor network.

I. INTRODUCTION

WSNs originate from applications in the military field and have characteristics of self-organization, high fault-tolerance, wide coverage, high detection accuracy and so on. Now it has been widely used in agriculture, forestry monitoring, medical and health fields [1], [2]. Compared with traditional wireless communication networks that focus on the quality of wireless communication service, WSNs concentrate on how to improve energy efficiency due to its limited energy and bandwidth resources [3]. As the scale of the WSN becomes larger and larger, the energy consumption of the traditional client/server computing model has increased and the delay grows longer due to the centralized data processing by sensor nodes. Another important aspect of constraint is that sensor nodes have limited energy and cannot deal with multiple applications. To solve these problems, Chen et al. [4] proposed a mobile-agent-based WSN model for reducing the energy consumption in a planar sensor network architecture.

The concept of the mobile agent was first proposed by General Magica in a commercial mobile agent system called Telescript. In WSNs, the mobile agent is allowed to carry processing codes for a specific task. For instance, the mobile agent can be a vehicular terminal equipped with communication and computing power [5]. It automatically migrates from one sensor node to another sensor node according to an established route and performs a specified task on the target node. Reference [6] proposed a route optimization method based on reinforcement learning. This kind of WSNs is efficient and robust for collaborative signal and information processing among multiple sensor nodes. Another significant advantage of the mobile agent is that it reduces the risk of network crashes when some of the sensor nodes are out of power.

Current researches on mobile-agent-based WSNs mainly focus on routing path planning. Several researchers designed their route algorithms based on clustering approaches for efficient data aggregation [7], [8]. However, most of these studies have suffered from an unbalanced load, large delays, and weak reliability [9]. A considerable amount of literature adopts itinerary planning approaches based on some classic route protocol [10], [11]. A major problem with this kind of application is the lack of intelligence [12].

Intelligence refers to the degree to which one can understand its own internal state and external environment, which is mainly reflected in three aspects: reaction, adaptation, and active capability [13]. An agent can react means that it has an autonomous choice to execute an action when it wants to. The adaptation implies the ability to learn. An active agent

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

can decide if it is necessary to adopt new or different intentional actions by recognizing the need to achieve goals.

With many successful applications of artificial intelligence in various fields, such as UAV scheduling [14], [15], Smart city services [16], [17], wireless sensor network [18], emotion recognition [19], [20], healthcare system [21]–[24], applying AI techniques to mobile agents has become a new idea and has a promising application prospect. Reference [25] improved the Q-learning algorithm to overcome the dimensionality problem in a multi-agent system. Reference [26] reviewed recent studies in mobile-agent-based healthcare to provide a deep insight into future applications. And wireless communication technology based on edge computing is proposed in [27]. [28] presented a mobile-agent-based system for assigning different modular robots to a fixed number of tasks using the ant colony algorithm. Reference [29] has developed a framework for information fusion and estimation over distributed multi-agent networks, which can improve the perception and intelligence of mobile agents. While all of the above work is creative, there is still a long way to go from AI.

The definition of artificial intelligence has been the subject of intense debate for decades of years. Computers are faster and more accurate than human brains, but few people say that computers have more intelligence than humans. Since the Industrial Revolution, human beings have been thinking about this problem. In 1948, Alan Mathison Turing outlined the field of artificial intelligence for the first time [30]. He conceived an abstract device called Turing Machine, which could perform calculations that any human mathematician could perform with the aid of algorithms in infinite time, power consumption, paper and pen, and perfect concentration. Artificial intelligence refers to more than the computing ability, which needs more information about world cognition and autonomy of choice [31].

With the development of artificial intelligence technology, many problems that need to be solved urgently can be optimized and solved [32]–[34]. In WSNs, mobile agents correspond to agents in AI. Agents constantly perform actions to receive the status and rewards from the environment. Mobile agents moving in the environment will inevitably affect the state of the external environment. Therefore, mobile agents need to change their strategies in real time. This paper combines the two, called iAgent, to improve the intelligence of the mobile agent. The key contributions of this article can be listed as follows.

1) Artificial intelligence technology is applied to mobile-agent-based wireless sensor networks, and an iAgent framework is proposed to increase the intelligence of mobile agents in routing. We design four iAgents in wireless sensor networks.
2) A new multi-iAgent algorithm is applied to reduce the energy consumption of the mobile-agent-based WSN and improve the lifetime of the wireless sensor network.
3) We conduct a simulation experiment to compare the performance of four iAgents and energy-efficiency of our proposed multi-iAgent algorithm.

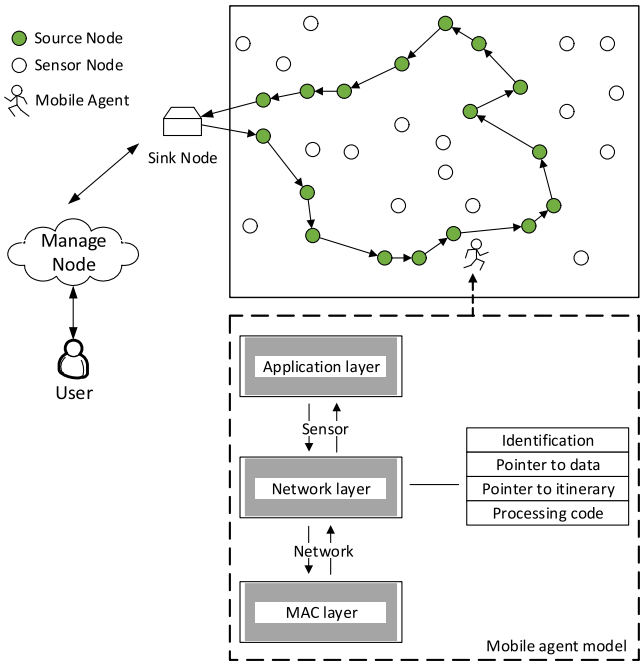

FIGURE 1. Mobile-agent-based wireless sensor network.

The remaining part of this paper is arranged as follows. Section II provides a brief introduction of the mobile agent in WSNs and then goes on to four types of iAgents when mobile agent combines with AI. Section III proposes a new multi-iAgent algorithm. Section IV is the experimental part. Four iAgents are compared and the multi-iAgent algorithm is demonstrated to be effective.

II. DESIGNS OF IAGENT

Sensor nodes, sink nodes, management nodes and mobile agents form a mobile-agent-based WSN by self-organization, as shown in Fig 1. A mobile agent starts from the sink node, migrates autonomously in the WSN to collect data and returns to the original sink node finally. The routing algorithm of the mobile agent is implemented in the network layer. The structure of a mobile agent has four attributes, as shown in the lower right corner of Fig 1. Identification is unique for each mobile agent. The data part stores the data the mobile agent collects from source nodes. Itinerary refers to the route path, which is the most important part. Different mobile agents carry different processing codes to finish different tasks, which are written before a mobile agent is sent out.

The main reason that mobile-agent-based WSNs are more efficient is that mobile agents have the ability to plan routes. Source nodes store the collected data locally. When mobile nodes reach the communication range of sensor nodes, sensor nodes transmit data to mobile nodes without using intermediate forwarding nodes. This can minimize the energy consumption of data forwarding and ensure the reliability of data. In this case, the main problem is to get the optimal trajectory of mobile agents to ensure coverage of all source nodes.

Algorithm 1 Simple-Reflex iAgent

Initialize state-action rules
for t = 1,2,...,N **do**
  perceive current state s_t
  get action a_t of s_t by state-action rules
end for

WSNs are designed to monitor the physical environment and gather the information that users are interested in. The limited capacity and power supply of sensor nodes make it unsuitable for WSNs to be a general service network system. Plus the strict real-time requirements of monitoring applications, WSNs must be closely coupled with specific applications in order to design an efficient application system. Completing different networks requires dispatching different agents without changing the procedures of sensor nodes. Aiming at different application requirements and combining artificial intelligence [35], we proposed four different agent programs for the mobile agent in WSNs to realize the mapping from perception to action.

A. SIMPLE-REFLEX IAGENT

This is the simplest iAgent, usually used in small-scale WSNs. The tasks issued by sink nodes are so easy that the action of iAgent can be decided by a series of rules. These rules have to do with the changes in WSNs, thus restricting the scale of WSNs. The simple-reflex iAgent can observe the outside environment and make a decision based on the current state. For example, the rule can be that when the iAgent arrives at a source node and finishes its task, it will search the nearest unvisited source node as the next node. Once the iAgent perceives that it has arrived at an unvisited source node, it will start to carry out the collecting job. Before the job is done, the state is set to have no impact on the iAgent. That is to say, as soon as the iAgent completes the task, it will look around and update its current state. The program is shown in Algorithm 1.

The route of the simple-reflex iAgent is fixed, predictable and controllable, thus resulting in more stable WSNs. However, this kind of iAgent only considers the current state, which will easily lead to many problems such as data load imbalance, delay, and security. The other disadvantage is that the iAgent moves along specified lines, leading to high energy consumption in local areas of the WSN. When some nodes have problems, it may even fall into endless circles.

B. MODEL-BASED IAGENT

The premise of simple-reflex iAgent is to be able to predict all possible changes in order to make rules, but this is impossible in practical applications. In WSNs, though the environment is not fully observable, the iAgent can enhance its perception by inference from knowledge base which stores the past experience. To improve the perception, multiple iAgents can also share their individual knowledge locally via existing self-organized communication technologies [36].

Algorithm 2 Model-Based iAgent

Initialize state-action rules, environment model E
for t = 1,2,...,N **do**
  perceive current state s_t
  get new state s'_t from E
  get action a_t of s'_t by state-action rules
  update E
end for

The inferred information can be divided into two parts. One is about how the world changes in the absence of the iAgent. For example, source nodes will generate data by observing the outside physical environment ceaselessly and weather may affect the quality of data. These are all unmeasurable factors and can only be obtained by rational thinking. The other part is the influence of the iAgent on the environment. The iAgent needs to know what the world will be once it takes action. If it moves to the node i and collects data, the environment will become that the node i is marked visited. Overall, an environment model is established to get these inferences from the knowledge base and then an action is decided based on the output of the model. The iAgent which has the environment model is called model-based iAgent, as shown in Algorithm 2.

Model-based iAgent can not only perceive the real-time information of monitoring nodes but also learn from the past experience by the environment model. For example, the density of sensor nodes will affect the quality of communication in WSN, when iAgent passes through congested areas or areas with poor signals, it will automatically decelerate and prolong transmission time. Another example is to adjust the motion scheme according to the communication range of monitoring nodes which set the power of the communication module to different levels according to their residual energy. When the communication range of each node changes, the route of the iAgent can be adjusted in time. This method can ensure the integrity of data collection and reduce energy consumption.

C. GOAL-BASED IAGENT

Different tasks of different applications have different requirements for data acquisition. If periodic monitoring data such as temperature and humidity are needed, source nodes will temporarily store data locally to reduce communication energy consumption and prolong network life. For high real-time data, such as environmental detection, battlefield assessment, and other applications, it is necessary to collect data ceaselessly. Model-based iAgent cannot adjust to different tasks, so a goal-based agent is proposed, which sets a goal according to the tasks announced by sink nodes, such as minimizing energy consumption or gathering data from several nodes, as shown in Algorithm 3.

Compared with the model-based iAgent, the goal-based iAgent has added a goal. Even at the same state, the iAgent

Algorithm 3 Goal-Based iAgent

Initialize environment model E, action model A, goal G
for t = 1,2,...,N **do**
  perceive current state s_t
  get new state s'_t from E
  get candidate actions $\hat{a}_t$ of s'_t from A
  get action a_t from $\hat{a}_t$ by G
  update E, A
end for

Algorithm 4 Utility-Based iAgent

Initialize environment model E, action model A, utility model U
for t = 1,2,...,N **do**
  perceive current state s_t
  get new state s'_t from E
  get candidate actions $\hat{a}_t$ of s'_t from A
  get action a_t from $\hat{a}_t$ by U
  update E, A, U
end for

can make different choices according to different targets, a goal can guide the iAgent toward the correct direction. Supposing the goal is to traverse all source nodes and collect data, the iAgent will tend to move to as many unvisited source nodes as possible. If the goal is changed to minimize energy consumption, then the iAgent may learn to choose the action that minimizes energy consumption.

Another difference to the previous iAgent is the way that the goal-based iAgent chooses an action. The simple-reflex iAgent and the model-based iAgent follow the established rules, while the goal-based iAgent has an action model. The action model is built to ensure that the iAgent has optional choices. Combined with the goal, the goal-based iAgent can choose the most suitable action from the available actions to achieve the goal. The objective is optimized by updating the environment model and the action model. Although goal-based iAgent is sometimes inefficient, it can increase flexibility which means that there is no need to change the code for different tasks. At this stage, search and plan are commonly used.

D. UTILITY-BASED IAGENT

The iAgent in WSNs often carries more than one goal to achieve. For example, sometimes they not only need to reduce energy consumption but also hope to collect more data, and safety is a considerable problem. It is common that these goals are interrelated. When there are conflicting goals, we weight these goals together as a utility. The aim of this iAgent is to maximize the utility of the goals.

The utility-based iAgent is similar to the goal-based iAgent except for the ways of choosing actions. The utility-based iAgent concentrates on the utility which reflects the performance of the iAgent more comprehensively. If all the goals are consistent and coordinate with each other, the performance will be improved greatly. The utility function can also be learned like the action model and the environment model. The iAgent updates the utility model to understand what benefit it will get in such a state if it takes some action.

In WSNs, there are various types of data that need to be collected in the same monitoring range. Therefore, iAgents in the network will undertake a variety of monitoring tasks. Obviously, different monitoring objects have different requirements for real-time data transmission. In multi-task WSNs, the usage of sensor networks can be changed by reallocating targets. If the data collected by sensor nodes need an immediate response, such as fire warning, the data will be sent back to sink through the real-time routing in the node routing table to meet the real-time requirements of the network. The algorithm of utility-based iAgent is shown in Algorithm 4.

We summarize these four kinds of iAgents for selecting actions, as shown in Fig 2. What's in the gray frame is the iAgent. The four iAgents interact with the environment. All iAgents have sensors and actuators that are physical components. They perceive the environment by sensors and make actions by actuators. Different colors represent different iAgents. The black line is the common flow that all iAgents will go through except for the part of the knowledge base and the environment model which are not possessed by the simple-reflex iAgent.

Intuitively, the simple-reflex iAgent is easiest to implement but the hardest to apply in real life for making rules that map states to actions is not an easy task. The model-based iAgent makes up for this deficiency and builds an environment model to reason the unknown information. In addition to the environment, the goal of the iAgent is critical to assist in completing tasks. The goal-based iAgent acts to achieve the goal. The one-goal situation is rare, most occasions have multiple goals, thus the utility-based iAgent is proposed to balance these goals so that the utility can be maximized.

Each of the four iAgent has the ability to learn and can choose the best actions through learning. The environment model, the action model, and the utility model are the components that can learn from the knowledge base. It is hard to conclude which is the best since only a specific application has the best iAgent.

III. MULTIPLE IAGENTS

Though the iAgent achieves good performs in WSNs, when the network scale becomes larger, some problems have arisen as below.

- The iAgent has to visit all source nodes in one round, which increases the data delay.
- Data load is unbalanced. Sensor nodes located at the end of the path need more power to receive and transmit because the size of the mobile agent is larger than that at the start of the path.
- It is hard to guarantee security. When the gathered data

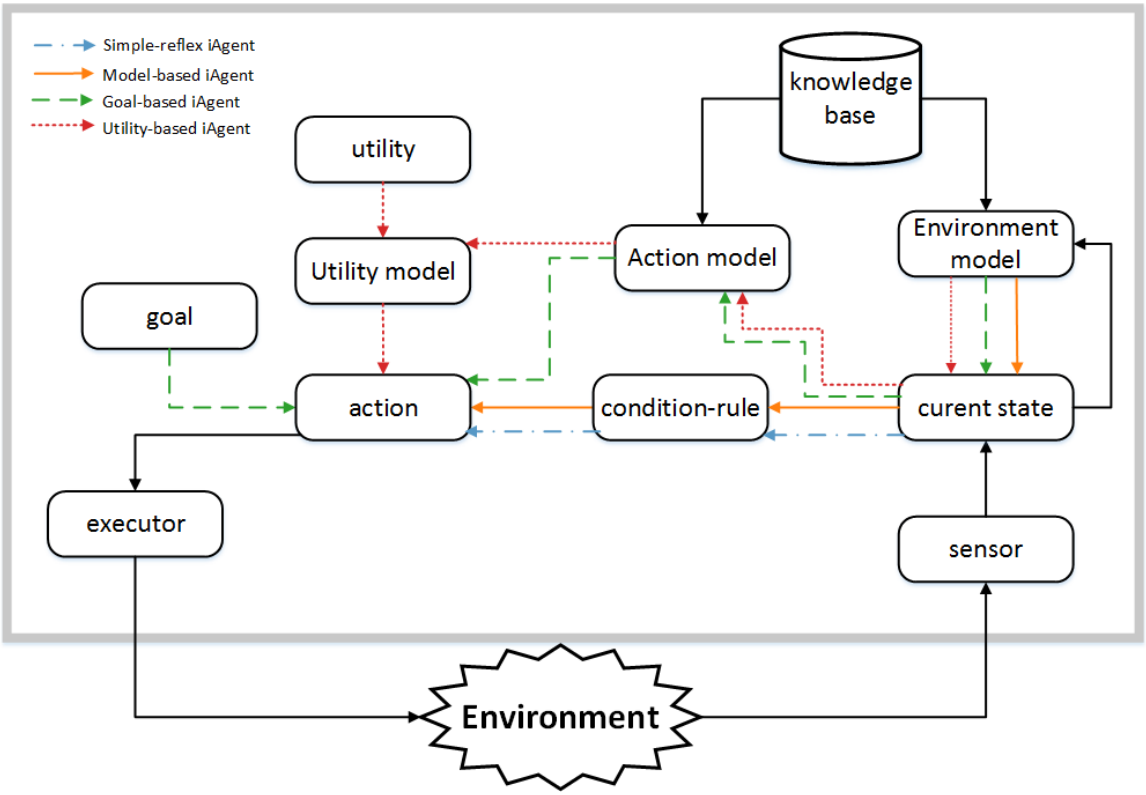

FIGURE 2. Four designs of iAgent.

become bigger and bigger, the cost to ensure safety has increased.

Taking these problems into consideration, most iAgent-based WSNs are designed to be multi-iAgent-based WSNs. In multi-iAgent-based WSNs, each iAgent carries a task, starts from the sink node, accesses a series of source nodes that are assigned to the iAgent before the trip, and returns to the starting point eventually. It can also be regarded as an iterative version of single-iAgent-based WSNs. It is usually divided into the following steps.

1) Determine the number of iAgents N.
2) Divide the source nodes into N groups.
3) Assign the groups to corresponding iAgents.
4) Determine the visiting order of the source nodes for each iAgent.

The last step can be regarded as the routing procedure of single iAgent as mentioned before. We can choose different types of iAgents to execute different tasks. The third step is taken after the source nodes are grouped. For the first and second steps, we propose a method to decide the number of iAgents and a cluster-based algorithm to divide the areas into several regions for iAgents.

The number of the iAgents is restricted to many factors such as the iAgents' physical capabilities, the size of the deploy area, the number of the source nodes, the size of the generated data by source nodes, the communication quality and so on. Here we choose several key elements $x_1, x_2, \ldots, x_m$ to build a linear model that outputs the number of the iAgents. The features of the model are designed according to a specific situation. The linear model tries to learn a linear combination of the features to predict the number of the iAgents, that is,

$$L = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_m x_m \tag{1}$$

Therein $\alpha_1, \alpha_2, \ldots, \alpha_m$ represent the importance of the attributes in prediction. α_0 is added as bias. These parameters can be estimated by the least square method.

The training data are produced by many past experiences. Before the iAgent sets out from the sink node, the linear model has been trained for several iterations and has achieved a certain degree of reliability. After the iAgent returns to the sink node, its itinerary can be added to the training materials to update the model.

The second step is to divide the source nodes into N groups. The main criterion for partition is the angle θ, as depicted in Fig 3. Establishing a coordinate system, what we want to get for each source node is the angle between the y-axis and a straight line connecting the sensor node and the source node. Then we can sort the source nodes by small to large order of the angle. Source nodes with similar angles are grouped together no matter how far they are. Once the groups of source nodes are determined, we can assign source nodes in the same group to the same iAgent.

It is expected that after collecting data from distant nodes, the path can be shorter so that less energy is spent on the

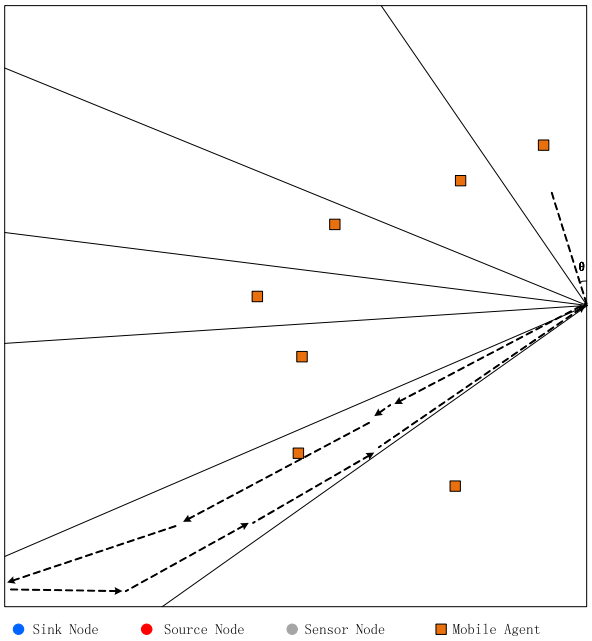

FIGURE 3. Multi-iAgent wireless sensor network.

Algorithm 5 Multiple Iagent Algorithm

Input sink node (x_0, y_0), source nodes
for $i = 1, 2, \ldots, M$ **do**
$\Delta x = x_i - x_0$
$\Delta y = y_i - y_0$
$\theta_i = arccos(\frac{\Delta x}{\sqrt{\Delta x^2 + \Delta y^2}})$
end for
sort the source nodes by θ_i
train a linear model $L = \sum_i^m \alpha_i x_i$
get the number of iAgent N from L
for $i = 0, 1, 2, \ldots, N-1$ **do**
choose source nodes $i \times \lfloor \frac{M}{N} \rfloor, \ldots, (i+1) \times \lfloor \frac{M}{N} \rfloor$
perform single-iAgent algorithm
end for

road. It means that the routing path from the sink node to the farthest node is preferably a straight line. Thus nodes with similar angles tend to be grouped together.

The algorithm combining the first step and the second step is as follows.

IV. SIMULATION AND PERFORMANCE EVALUATION

We have established a simulation system to simulate the WSN environment. They are made up of a set of sensor nodes, a sink node, and multiple iAgents. The purpose of our experiment is to demonstrate the energy-effective performance of our proposed methods. There are two groups of experiments. One is to compare four single iAgents including simple-reflex iAgent, model-based iAgent, goal-based iAgent, and utility-based iAgent. The other is to show the advantages of the multi-iAgent system over the single-iAgent system.

TABLE 1. The main parameter settings.

Parameters	Values
Simulation area	500 m × 500m
Number of sensor nodes	1000
Number of source nodes(default)	40
Coordinate of sink node	(505, 250)
Radio range	100m
Mobile agent size	1K bytes
Size of sensed data packet(default)	4K bytes
Data compression ratio	0.45
Transmission energy	50nJ/bit
Receiving energy	50 nJ/bit
Transmitter Amplifier Efs	$10pJ/bit/m^2$
Transmitter Amplifier Emp	$0.0013pJ/bit/m^4$

A. SIMULATION SETTING

Due to the high cost and difficulty of deploying large-scale WSNs, most of the current research work on WSNs is carried out in the simulation environment. In our experiment, the following assumptions are made.

- The sensor nodes are fixed all the time.
- Each node has the same communication capability and initial energy and knows its location information.
- The sink node knows the information of source nodes.
- The sensor nodes are independent of each other.
- The iAgents act independently and do not influence each other.
- There are no overlapping sensor nodes.

The experiments are conducted in a square area of 500 $m \times 500m$. The sink node is situated in the right center edge of the monitoring region. The sensor nodes are randomly deployed before the simulation begins and do not move during the experiment. Each source sensor generates the same amount of data.

The iAgent has localization function, so it knows where it is. It gathers data generated by sensor nodes in a certain range rather than collecting data from the environment. Starting from the sink node, the iAgent collects data from source nodes by the multi-hop method. Eventually, the iAgent will come back to the base node to hand the data to the sink node. When it is at the sink node, it can supplement energy, so the energy is regarded as infinite. The initial size of a mobile agent is 1K bytes. The parameters are as summarized in Table 1.

We use energy consumption as a metric to compare different methods. The energy consumption refers to the sum of energy consumed during transmission and receiving in one round. In the multi-iAgent environment, the energy consumption of each iAgent is added together as the total energy consumption. Since the sink node is thought to have a sufficient energy supply, the energy consumption of the sink node can be ignored. The WSN is affected by many factors, we choose the number of source nodes and the size of the sensed data to observe their effect on energy consumption. The number

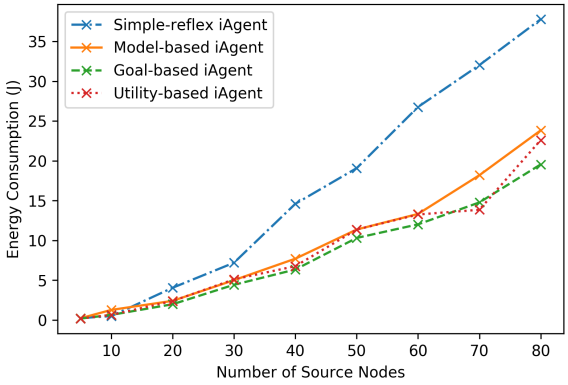

FIGURE 4. Comparison of energy consumption by the number of source nodes.

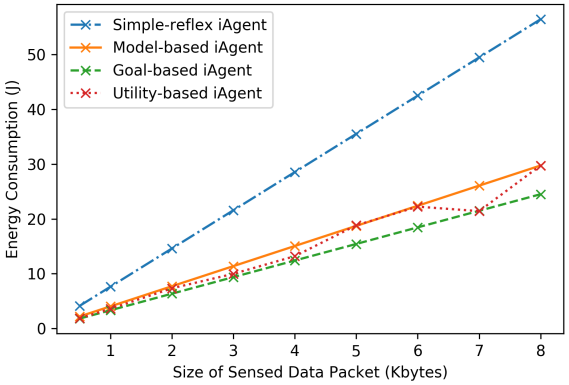

FIGURE 5. Comparison of energy consumption by the size of sensed data.

of source nodes is increased to show the trend of energy consumption. We also change the size of the sensed data packets at each sensor to make comparisons. To make the results more intuitive, two routing diagrams of the single-iAgent method and the multi-iAgent method are plotted.

B. COMPARISON OF FOUR IAGENTS

In this section, we compare the energy consumption of the four iAgents by different numbers of source nodes and different sizes of the sensed data packet. There are at least 5 source nodes, then 10 source nodes, and from 10 source nodes, we increase to 80 source nodes in 10 intervals. The size of the sensed data packet starts at 0.5K bytes, then increases to 1K bytes, and then increases by the step size of 1kbytes. The largest size of the packet can be 8K bytes. The experiment results are shown in Fig 4 and Fig 5.

From the above two figures, we can infer that the simple-reflex iAgent consumes the most energy. The performance of the model-based iAgent is at the medium level of the four iAgents. The goal-based iAgent outperforms other iAgents in most situations, probably because that the goal-based iAgent targets at minimizing the energy consumed. The energy consumption is one part of the objects of the utility-based iAgent, it is not devoted to minimizing energy consumption. What's more, the utility-based iAgent is easily affected by other things, this type of iAgent may be unstable in terms of energy consumption.

In Fig 4, energy consumption increases as the number of source nodes increases. When the number of source nodes is lower than 30, the performance of the four iAgents is comparable. But when the number of source nodes exceeds 30, the energy consumption of the simple-reflex iAgent begins to increase sharply, and the other three methods are left far behind. This is because the simple-reflex iAgent requires us to specify rules beforehand. However, as the source nodes increases, the network becomes more and more complex, it is difficult for us to observe all the cases, so this method begins to show deficiencies.

The trend of energy consumption in Fig 5 is consistent with that in Fig 4. As the amount of sensed data in the WSN increases, the energy consumption becomes larger and larger. Except for utility-based iAgent, the energy consumed by the other three iAgents is linearly related to the size of the perceived data packet. The goal-based iAgent can be said to be an improved version of model-based iAgent for it adds a goal to make iAgent act purposefully.

Although these four iAgents have obvious advantages and disadvantages in our simulation experiments, the model-based iAgent is often the first choice for many WSNs because of its simple and fast characteristics in practical application. The simple-reflex iAgent can have very good results when the state-action rules are carefully designed. From simple-reflex iAgent, model-based iAgent, goal-based iAgent to utility-based iAgent, the complexity gradually increases. In practice, it is still necessary to choose the proper iAgent according to the specific scenario design requirements.

C. COMPARISON OF SINGLE-IAGENT SYSTEM AND MULTI-IAGENT SYSTEM

To demonstrate the performance of the multi-iAgent algorithm, we compare it with the single-iAgent algorithm. To study the impact of the number of source nodes on energy consumption, We increase the number of source nodes from 10 to 80 by steps of 10 under the condition that the size of the sensed data packet is 4K bytes. When the source nodes are fixed, we change the size of the sensed data to compare two algorithms. The results are shown in Fig 6 and Fig 7.

In Fig 6, it is obvious and normal that when the number of source nodes increases, the energy consumption grows higher. When the number of source nodes is fixed, the single-iAgent always performs better than multi-iAgent except when the number of source nodes is below 10. Inferring from the multi-iAgent algorithm, we can find that when the number of source nodes is below 10, the number of iAgents determined by the multi-iAgent algorithm is 1, so the performance of the two are the same. At first, the gap between the two sides was not obvious. But when source nodes increase, the gap grows larger. The reason may be that the accumulated energy will cost more energy.

In Fig 7, the multi-iAgent always consumes less energy than the single-iAgent. Even when the sensor data is

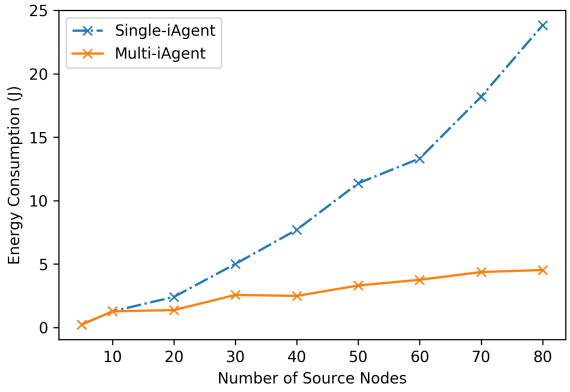

FIGURE 6. Comparison of energy consumption by the number of source nodes.

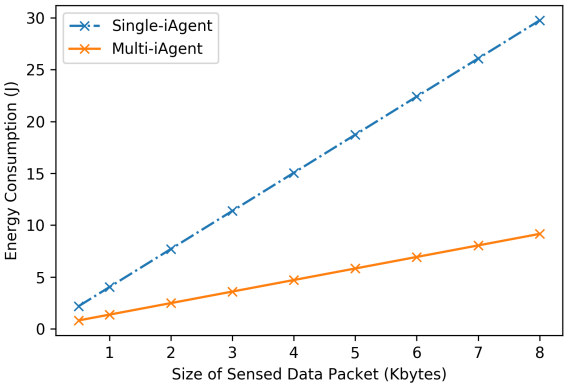

FIGURE 7. Comparison of energy consumption by the size of sensed data.

very small, the energy consumption of multi-agent is obviously larger than that of single-iAgent, and the gap between them is widening with the increase of sensor data.

D. ROUTING DIAGRAMS

To make it clear, we plot the routing diagrams of the two algorithms, as depicted in Fig 8 and Fig 9. The number of sources node is 40 and the size of the sensed data for each source node is 4K bytes. The right blue point represents the sink node, and there are 1000 sensor nodes in gray color in the figure. The red ones are source nodes. Each iAgent starts from the sink node, collects data from the source nodes and finally returns to the sink node.

The energy consumption for the single-iAgent WSN and the multi-iAgent WSN are 15.1J and 4.7J, respectively. Not only the energy consumption of the single-iAgent is higher than that of the multi-iAgent, but the sum of the routing lengths in the single-iAgent WSN is also longer than that in the multi-iAgent WSN. So it can be inferred that the routing length and the energy consumption have a positive correlation.

Due to the limited energy of sensor nodes, energy saving is the most important task of WSNs. From the above routing diagram, we can reduce energy consumption by reducing the length of the routing, or we can send an appropriate number of iAgents to save energy.

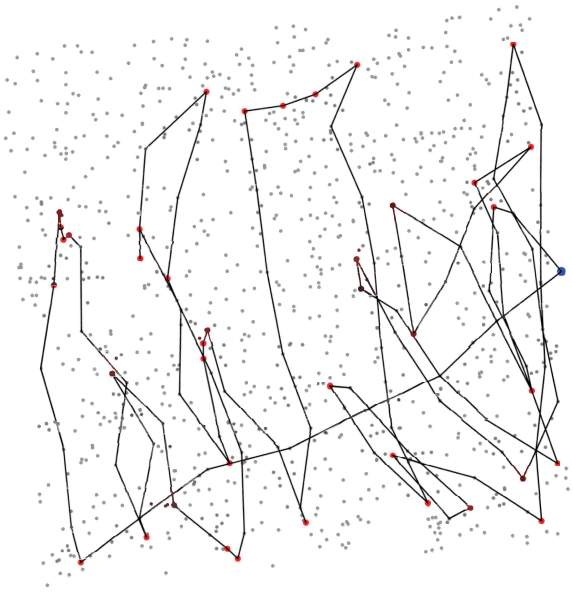

FIGURE 8. Routing diagram of single-iAgent WSN.

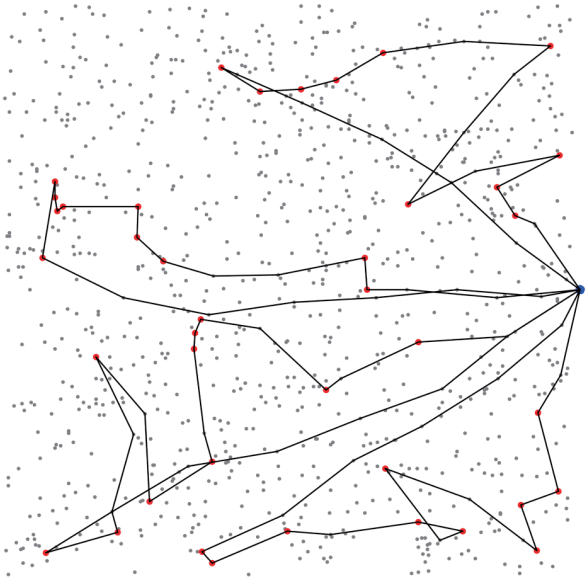

FIGURE 9. Routing diagram of multi-iAgent WSN.

V. CONCLUSION

In this paper, we have discussed the development and application of mobile agents in WSNs and proposed four designs of iAgent combined with artificial intelligence. Due to the limit of the single-iAgent WSN, we also proposed a new multi-iAgent WSN. The experiments show that the goal-based iAgent performs best with respect to energy consumption. The multi-iAgent WSN is demonstrated to perform better than the single-iAgent WSN. Many experiments show that these methods can save significant energy and thus significantly reduces costs on a large scale.

In the near future, we will expand our work from three aspects. The first is security, which is an inevitable problem in mobile agents [37]–[39]. We will study how to ensure the

security of data carried by the iAgents. The second problem is about the interaction between iAgents [40]. In the simulation environment, we assume that iAgents are not related to each other, but it certainly will not be so in practical application. What's more, the combination of mobile agents and AI can bring good benefits, but the lack of exchange of information between iAgents will lead to loss of information. We will study the interaction of the iAgents in the multi-iAgent environment in the next step. The last one is about the mobility of the iAgents. To simplify the problem, we assume that the sensor nodes in the WSNs are static while the sensor nodes in some WSNs can move. To generalize our model, we will abandon this simulation hypothesis and conduct deep research on mobility [41].

REFERENCES

[1] C. Wang, H. Lin, and H. Jiang, "CANS: Towards congestion-adaptive and small stretch emergency navigation with wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 5, pp. 1077–1089, May 2016.
[2] H. Jiang, C. Cai, X. Ma, "Smart home based on WiFi sensing: A survey," *IEEE Access*, vol. 6, pp. 13317–13325, 2018.
[3] M. Ezhilarasi and V. Krishnaveni, "A survey on wireless sensor network: Energy and lifetime perspective," *Taga J. Graphic Technol.*, vol. 14, pp. 3099–3113, Apr. 2018.
[4] M. Chen, T. Kwon, Y. Yuan, and V. C. M. Leung, "Mobile agent based wireless sensor networks," *J. Comput.*, vol. 1, no. 1, pp. 14–21, 2006.
[5] J. Zhou, D. Tian, Y. Wang, Z. Sheng, X. Duan, and V. C. M. Leung, "Reliability-optimal cooperative communication and computing in connected vehicle systems," *J. Comput.*, to be published. doi: 10.1109/TMC.2019.2907491.
[6] P. Sun, Y. Hu, J. Lan, L. Tian, and M. Chen, "TIDE: Time-relevant Deep Reinforcement Learning for Routing Optimization," *Future Gener. Comput. Syst.*, vol. 99, pp. 401–409, Oct. 2019. doi: 10.1016/j.future.2019.04.014.
[7] S. Sasirekha and S. Swamynathan, "Cluster-chain mobile agent routing algorithm for efficient data aggregation in wireless sensor network," *J. Commun. Netw.*, vol. 19, no. 4, pp. 392–401, 2017.
[8] N. Y. SreeRanjani, A. G. Ananth, and L. S. Reddy, "An energy efficient data gathering scheme in wireless sensor networks using adaptive optimization algorithm," *J. Comput. Theor. Nanosci.*, vol. 15, nos. 11–12, pp. 3456–3461, 2018.
[9] Y. Zhang, H. Wen, F. Qiu, Z. Wang, and H. Abbas, "iBike: Intelligent public bicycle services assisted by data analytics," *Future Gener. Comput. Syst.*, vol. 95, pp. 187–197, Jun. 2019.
[10] J. Wang, Y. Zhang, Z. Cheng, and X. Zhu, "EMIP: Energy-efficient itinerary planning for multiple mobile agents in wireless sensor network," *Telecommun. Syst.*, vol. 62, no. 1, pp. 93–100, May 2016.
[11] K. Lingaraj, R. V. Biradar, and V. C. Patil, "OMMIP: An optimized multiple mobile agents itinerary planning for wireless sensor networks," *J. Inf. Optim. Sci.*, vol. 38, no. 6, pp. 1067–1076, 2017.
[12] Y. Zhang, X. Ma, J. Zhang, M. S. Hossain, G. Muhammad, and S. U. Amin, "Edge intelligence in the cognitive Internet of Things: Improving sensitivity and interactivity," *IEEE Netw.*, vol. 33, no. 3, pp. 58–64, May/Jun. 2019.
[13] M. Chen, F. Herrera, and K. Hwang, "Cognitive computing: Architecture, technologies and intelligent applications," *IEEE Access*, vol. 6, pp. 19774–19783, 2018.
[14] J. Yang, X. You, G. Wu, M. M. Hassan, A. Almogren, and J. Guna, "Application of reinforcement learning in UAV cluster task scheduling," *Future Gener. Comput. Syst.*, vol. 95, no. 1, pp. 140–148, 2019.
[15] Y. Zhao, J. Ma, X. Li, and J. Zhang, "Saliency detection and deep learning-based wildfire identification in UAV imagery," *Sensors*, vol. 18, no. 3, p. 712, 2018.
[16] S. Xiao, H. Yu, Y. Wu, Z. Peng, and Y. Zhang, "Self-evolving trading strategy integrating Internet of Things and big data," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2518–2525, Aug. 2018.
[17] M. Chen and Y. Hao, "Label-less learning for emotion cognition," *IEEE Trans. Neural Netw. Learn. Syst.*, 2019. doi: 10.1109/TNNLS.2019.2929071.
[18] J. Lu, L. Feng, J. Yang, M. M. Hassan, A. Alelaiwi, and I. Humar, "Artificial agent: The fusion of artificial intelligence and a mobile agent for energy-efficient traffic control in wireless sensor networks," *Future Gener. Comput. Syst.*, vol. 95, pp. 45–51, Jun. 2019.
[19] Y. Qian, Y. Zhang, Y. Ma, H. Yu, and L. Peng, "EARS: Emotion-aware recommender system based on hybrid information fusion," *Inf. Fusion*, vol. 46, pp. 141–146, Mar. 2019.
[20] M. S. Hossain and G. Muhammad, "Emotion recognition using deep learning approach from audio–visual emotional big data," *Inf. Fusion*, vol. 49, pp. 69–78, Sep. 2019.
[21] M. Chen, W. Li, Y. Hao, Y. Qian, and I. Humar, "Edge cognitive computing based smart healthcare system," *Future Gener. Comput. Syst.*, vol. 86, pp. 403–411, Sep. 2018.
[22] M. S. Hossain, G. Muhammad, and A. Alamri, "Smart healthcare monitoring: A voice pathology detection paradigm for smart cities," in *Multimedia Systems*. Berlin, Germany: Springer-Verlag, 2017. doi: 10.1007/s00530-017-0561-x.
[23] M. S. Hossain, S. U. Amin, G. Muhammad, and M. Al Sulaiman, "Applying deep learning for epilepsy seizure detection and brain mapping visualization," *Future Gener. Comput. Syst.*, vol. 86, pp. 403–411, Sep. 2018.
[24] Y. Zhang, R. Gravina, H. Lu, M. Villari, and G. Fortino, "PEA: Parallel electrocardiogram-based authentication for smart healthcare systems," *J. Netw. Comput. Appl.*, vol. 117, pp. 10–16, Sep. 2018.
[25] D. Luviano-Cruz, F. Garcia-Luna, L. Pérez-Domínguez, and S. K. Gadi, "Multi-agent reinforcement learning using linear fuzzy model applied to cooperative mobile robots," *Symmetry*, vol. 10, no. 10, p. 461, 2018.
[26] S. Iqbal, W. Altaf, M. Aslam, W. Mahmood, and M. U. G. Khan, "Application of intelligent agents in health-care," *Artif. Intell. Rev.*, vol. 46, no. 1, pp. 83–112, 2016.
[27] D. Wang, Y. Peng, X. Ma, W. Ding, H. Jiang, F. Chen, and J. Liu, "Adaptive wireless video streaming based on edge computing: Opportunities and approaches," *IEEE Trans. Services Comput.*, to be published. doi: 10.1109/TSC.2018.2828426.
[28] B. Qian and H. H. Cheng, "A mobile agent-based coalition formation system for multi-robot systems," in *Proc. 12th IEEE/ASME Int. Conf. Mechtron. Embedded Syst. Appl. (MESA)*, Aug. 2016, pp. 1–6.
[29] D. Tian, J. Zhou, and Z. Sheng, "An adaptive fusion strategy for distributed information estimation over cooperative multi-agent networks," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3076–3091, May 2017.
[30] A. M. Turing, "Computing machinery and intelligence," in *Parsing the Turing Test*. Dordrecht, The Netherlands: Springer, 2009, pp. 23–65.
[31] M. Chen and V. Leung, "From cloud-based communications to cognition-based communications: A computing perspective," *Comput. Commun.*, vol. 128, pp. 74–79, Sep. 2018.
[32] M. Chen, Y. Hao, H. Gharavi, and V. Leung, "Cognitive information measurements: A new perspective," *Inf. Sci.*, 2019. [Online]. Available: https://arxiv.org/abs/1907.01719
[33] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mech. Syst. Signal Process.*, vol. 108, pp. 33–47, Aug. 2018.
[34] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu, and I. Humar, "A dynamic service migration mechanism in edge cognitive computing," *ACM Trans. Internet Technol.*, vol. 19, no. 2, P. 30, 2019.
[35] S. J. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, Malaysia: Pearson Education Limited, 2016.
[36] D. Tian, J. Zhou, Z. Sheng, M. Chen, Q. Ni, and V. C. M. Leung, "Self-organized relay selection for cooperative transmission in vehicular ad-hoc networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9534–9549, Oct. 2017.
[37] N. Borselius, "Mobile agent security," *Electron. Commun. Eng. J.*, vol. 14, no. 5, pp. 211–218, 2002.
[38] Y. Hedin and E. Moradian, "Security in multi-agent systems," *Procedia Comput. Sci.*, vol. 60, pp. 1604–1612, Dec. 2015.
[39] P. Zhao, J. Li, F. Zeng, F. Xiao, C. Wang, and H. Jiang, "ILLIA: Enabling k-anonymity-based privacy preserving against location injection attacks in continuous LBS queries," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1033–1042, Apr. 2018.
[40] M. M. Gulzar, S. T. H. Rizvi, M. Y. Javed, U. Munir, and H. Asif, "Multi-agent cooperative control consensus: A comparative review," *Electronics*, vol. 7, no. 2, p. 22, 2018.
[41] M. Chen, Y. Hao, C. Lai, D. Wu, Y. Li, and K. Hwang, "Opportunistic task scheduling over co-located clouds in mobile environment," *IEEE Trans. Service Comput.*, vol. 11, no. 3, pp. 549–561, May/Jun. 2018.