Archives of Computational Methods in Engineering (2021) 28:1323–1334
https://doi.org/10.1007/s11831-020-09417-1

ORIGINAL PAPER

A Survey on Mobile Agent Itinerary Planning for Information Fusion in Wireless Sensor Networks

Mohamed El Fissaoui[1] · Abderrahim Beni-hssane[1] · Slimane Ouhmad[2] · Khalid El Makkaoui[3]

Received: 10 July 2019 / Accepted: 24 February 2020 / Published online: 4 March 2020

Abstract
In wireless sensor networks (WSNs), mobile agent (MA) is a new paradigm that has gained more attention lately. In this paradigm, MA migrates among sensor nodes (SNs) for data fusion. However, in comparison to traditional client/server paradigm; where each SN in the network sends its collected data to the sink, MA presents a good alternative in terms of energy consumption, response time and network lifetime. In this paradigm, the most critical properties of MA is the itinerary planning, it has always been an issue and a NP-hard problem. In this paper, we present a survey of the proposed itinerary planning approaches for MAs in WSNs. MA itinerary planning techniques can be classified into three categories: static itinerary planning, dynamic itinerary planning and hybrid itinerary planing. The benefits and shortcomings of different MA itinerary planning approaches are presented. Furthermore, we implement, simulate and compare the most prominent itinerary planning algorithms upon a common parameter space, making realistic network-level assumptions.

1 Introduction

Recently, wireless sensor networks (WSNs) [2] have received great attention for wide application potentials. WSNs applications such as, battlefield surveillance, target tracking [44], home appliances and inventory tracking enlarge human capabilities to remotely interact with the physical world. In most of these applications, data fusion [19] is needed. WSN consists of a hundreds or thousands of autonomous sensor nodes (SNs) spatially distributed to collect data from the surrounding environment then send it back to the sink. WSNs also provide the possibility of collaborative information processing [48], which require a sensor network system to process data cooperatively and to combine information from multiple sources. Since data from multiple SNs with overlapping sensing regions is almost always correlated, one can remove the redundant information in the data, through in-network aggregation [13] and compression [41] local to the nodes that generate the data, before shipping it to a remote node.

In WSNs, the most commonly used paradigm for data fusion is client–server (CS); where the SNs send the collected data to the sink where the data processing take place [1]. As Fig. 1a shows, all SNs send the collected data to the sink via multi-hop routing [40]. There are some shortcomings with this paradigm [38]. In one hand, it requires many round trip to complete one transaction, also the network needs to be alive and healthy the entire time of the transaction. Thus consumes network resources such as bandwidth and energy, and decreasing the life time of the network. In the other hand, some super nodes with bigger storage and high computation capabilities need to be available in this model.

To overcome the aforementioned shortcomings of CS paradigm, a new computing paradigm have been proposed [38] based on mobile agent (MA). In this paradigm, instead of sending the collected data by the SNs to the

✉ Mohamed El Fissaoui
Elfissaoui.m@ucd.ac.ma

Abderrahim Beni-hssane
abenihssane@yahoo.fr

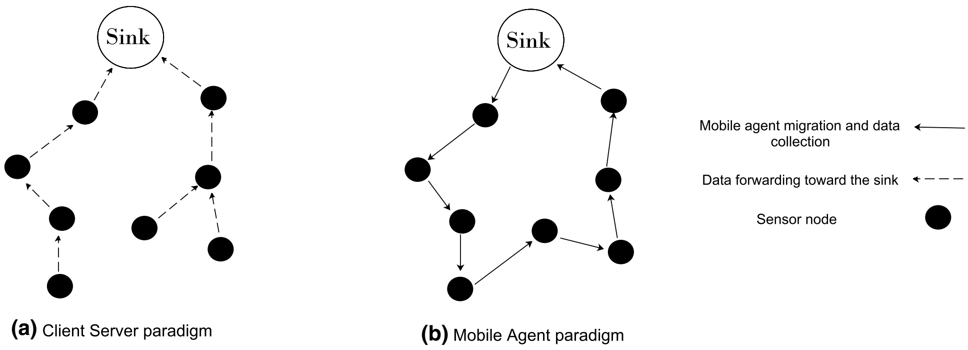
Slimane Ouhmad
ouhmad_slimane@yahoo.fr

Khalid El Makkaoui
kh.elmakkaoui@gmail.com

[1] LAROSERI Laboratory, FS, Univ Chouaib Doukkali, B.P. 20, 24000 El Jadida, Morocco

[2] LAVETE Laboratory, FST, Univ Hassan 1, B.P. 577, 26000 Settat, Morocco

[3] Laboratory of IPOSI, Mathematics and Computer Science Department, National School of Applied Team of Modeling and Scientific Computing, FP, Univ Mohammed 1, B.P. 300, 62700 Selouane, Nador, Morocco


Sink
Sink
Mobile agent migration and data collection
Data forwarding toward the sink
Sensor node

(a) Client Server paradigm

(b) Mobile Agent paradigm

Fig. 1 Data fusion paradigms in WSNs

sink, the processing code (MA) is moved to the data locations through MA for data fusions. Fig. 1b illustrates the MA paradigm. As the figure shows, the migration of MA starts from the sink and visits the SNs in its itinerary collecting data from these SNs, then the MA migrates back to the sink with the collected data. However, this paradigm has some disadvantages in some particular scenarios, such as security issues [43]. Figure 1 illustrates Client/Server and Mobile Agents Paradigms.

In [33] authors listed many features of this paradigm including : scalability, reliability, extensibility and task adaptivity, energy awareness, progressive accuracy. In comparison to the traditional data fusion model in WSNs, where individual SNs send the collected data to the sink, MA paradigm has been proven to be efficient in terms of energy consumption, network lifetime and over all time response [28, 46].To migrate amongst SNs, an itinerary should be planned for MA. An itinerary is a route that the MA follow when migrating amongst SNs. It has been proved that planning itinerary for MA is an NP-Hard problem [45] and the most challenging issue facing this paradigm.

The efficiency of MAs in the WSNs, make suitable for this kind of networks. Many applications adopt MAs based paradigm in WSN such as object detection/tracking [42, 48], healthcare [3], agriculture [14], control/assistant [20], security systems [32], medical/human-care systems, unmanned aerial vehicles, and mobile robots. The overall scheme of MA-based applications in WSNs is illustrated in Fig. 2.

The desire to design the itinerary for MA for data fusion in an energy efficient manner makes the use of MA paradigm in WSNs applications, largely, determined by the methods used to plan the itinerary for the MA [7]. The itinerary planning for MA can be classified into three main categories: static, dynamic or hybrid which can be classified in its turn in single and multiple.

In single static itinerary planning (SSIP), the sink collects the information of network then based on this information the MA itinerary is computed, the MA migration starts form the sink then next node in the itinerary and so on till the last SN in the network. SSIP is satisfactory for small to medium scale WSN.

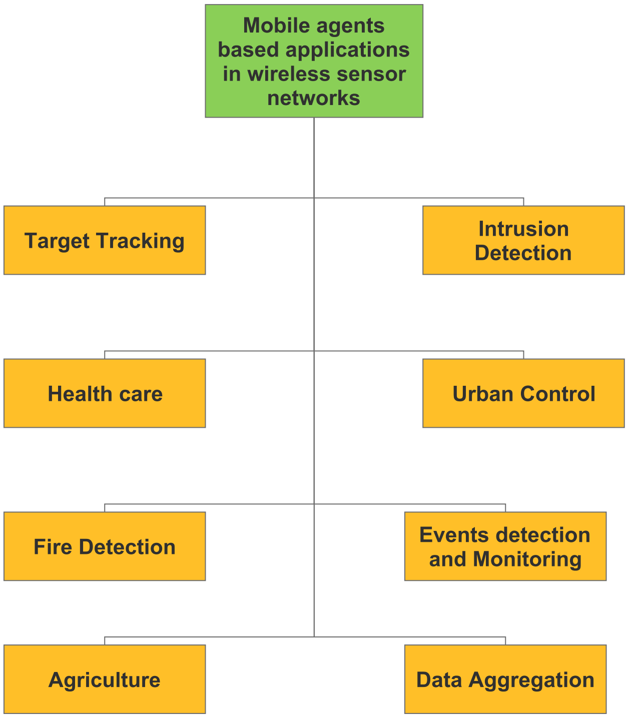


Fig. 2 Mobile Agents paradigm based application in the WSNs

Table 1 WSN application based itinerary planning for MAs

	Data fusion	Target tracking	localization	Intrusion detection
SIP	Yes	No	No	No
DIP	No	Yes	Yes	Yes
HIP	No	Yes	No	Yes

The type of the itinerary planning for MAs depends on the desired application, as Table 1 shows, SIP are suitable for data fusion application, DIP are used by target traking, intrusion detection, and localization applications. The HIP can be used for target traking, intrusion detection.

To bypass the lack of scalability in SSIP, multiple static itinerary planning (MSIP) present a good choice.In MSIP, the sink collect the global information of network topology and computes multiple itineraries or paths of the MAs by using the collected information ,then each all MA start its migration in parallel. Static itinerary planning (SIP) is more suitable for data monitoring applications.

Single dynamic itinerary planning (SDIP) allow more flexibility, deciding the next destination node on the fly, SDIP can avoid the node failure and choose another node to continue its migration, even the high cost of SDIP in terms of energy consumption and over all time response this scheme is more flexible in case of node failure. But the same as SSIP, SDIP suffer from lack of scalability with the growing size of the network.

SDIP became inefficient with the growing size of the network, therefore a multiple dynamic itinerary planning MDIP solution have been adopted by researchers. This approach allow more scalability, also due to the critical issue of energy consumption in WSN, the usage and parallel deployment of multiple mobile agents for data aggregation has been proven to be an efficient scheme in energy consumption cost and overall response time.

MDIP approach give the MAs the choice to decide the next destination node on the fly (based on residential energy, trust level...), it means that if there is any node failure, the MAs choose another destination node and so on, but also this approach may not be the best solution, because the more intelligent within the MAs the bigger the MAs are, which affect the energy and the overall time response.

MDIP allow more flexibility, even its high energy consumption an overall time response, as Fig. 2a shows, without node failure MDIP normally, and after node failure in Fig. 2 the MAs agent detect the failure of node and choose another node as its next destination node.

2 Mobile Agents Technology

Mobile agent technology is a paradigm for programming distributed applications over large networks such as wireless sensor networks (WSNs). The mobile agent is of great interest for implementing applications whose performance varies depending on the availability and quality of services and resources. As a result, mobile agents are an obvious choice in the field of wireless sensor networks. Several researches have proposed schemes to effectively use the mobile agent in this kind of networks. They have been found particularly useful for facilitating the fusion and efficient dissemination of data in sensor networks [7]. In mobile agent-based data fusion tasks, the choice of agent routes is of critical importance affecting the overall cost of energy consumption.

2.1 Mobile Agent Concept

The growth in user requirements and preferences, as well as the need for incremental scalability of applications, ensures that a centralized, rigid and passive vision reaches its limits. On the other hand, recent technological progress has led to the emergence of a new means facilitating the task of designing and developing these applications. Indeed, the rise of distributed computing has been accompanied by the emergence of the paradigm of mobile agents (MAs). The latter offers greater flexibility and treatment efficiency [12, 21, 22, 33]. Mobile agent technology presents interesting prospects for various application domains of the present era.

A mobile agent (MA) is a special kind of software that migrates among SNs to autonomously carry out task(s) in response to changing conditions in the network environment to achieve the objectives of the agent dispatcher [48].

2.2 Characteristics of a Mobile Agent

An agent is essentially characterized by the following points [Qih]:

- *Nature* The agent can be a physical or virtual entity; the physical entity is something that acts in the real world, for example: a robot, an airplane or a car.On the other hand, the virtual entity is an entity that does not exist physically and represents a software component or a computer module.
- *Autonomy* This means that it is not driven by commands from the user or another agent, but by itself.Therefore it has a certain freedom of movement.
- *The environment* it represents the space in which the agent is able to perceive and act.
- Communication: The agent has the ability to communicate directly with other agents.
- *Efficiency* it represents the speed of execution and intervention of the agent.
- R*epresentational capacity* Agents only have a partial representation of their environment, that is, they do not have a global vision of everything that happens to their environment. An agent does not know all the details, it has only three types of knowledge; domain knowledge, control knowledge, communication and interaction skills.
- *Own resources* To enable the agent to act in the environment, it needs a number of resources: energy, CPU, amount of memory, access to some sources of information. These resources make the agent dependent on his environment, but they give it a certain independence, being able to manage them.
- *The objective* The agent is thus a kind of “living organism” whose behavior, which is summarizes to communicate, to act and possibly to reproduce, aims at the

satisfaction of its needs and objectives from all other elements (perceptions, representations, actions, communications and resources).

There are other optional features that distinguish agents such as [30]:

- *The reasoning* the agent can be linked to an expert system or other mechanisms of more or less complex reasoning.
- *The control* it can be totally distributed between the agents, but can be dedicated to a certain class of agents as facilitator agents.
- *Anticipation* the agent can more or less have the ability to anticipate events future.
- *Granularity or complexity* the agent can be very simple like a neuron but also more complex.
- *The contribution* the agent participates more or less in solving the problem or in the activity overall system.
- *Intentionality* An intentional agent is an agent guided by his goals. An intention expresses the will of an agent to reach a goal or to perform an action.
- *Rationality* Rational agents have criteria for evaluating their actions, and select according to these criteria the best actions to reach the goal.
- *Adaptability* An adaptable agent is an agent able to control his abilities (communicational, behavioral, etc.) according to the environment.
- *Commitment* The notion of commitment is one of the essential qualities of cooperative agents. A cooperative agent plans its actions by coordination and negotiation with the other agents. By building a plan to achieve a goal, the agent gives himself the means to achieve it and therefore commits to accomplish the actions that satisfy this goal; the agent believes he has developed, which leads him to act accordingly.
- *Intelligence* An intelligent agent is a cognitive agent, rational, proactive and adaptive.

2.3 Mobility

Mobility can be referred to as physical mobility or software mobility. Physical mobility is the movement of a physical terminal, such as a laptop or phone. While software mobility is the movement of a software program between two or more physical terminals. The mobile software entity is then called component, agent, mobile application [35]. Figure 3 illustrates the mobility degree of a Mobile Agent.

2.3.1 Degree of Mobility

There are two degrees of mobility of applications: weak and strong [5]:

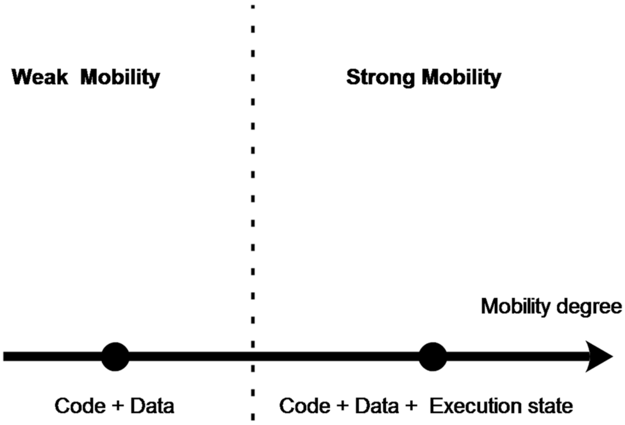


Fig. 3 Mobility degree

2.3.2 Weak Mobility

It consists of transferring only the data and code of an application that moves during its execution on a source machine to a destination machine. So, if we find an interruption in the execution of the application, the latter resumes its execution from the beginning, while having the updated values of its data.

2.3.3 Strong Mobility

It consists of transferring the data, the code plus the execution state of an application that is moving during its execution on a source machine to a destination machine. So, if we find an interruption in the execution of the application, the latter resumes its execution from the point where it was interrupted on the starting machine.

2.3.4 Resources Needed for Mobility

If a system supports weak or strong mobility, there are always specific resources that should emigrate with the agent depending on the degree of mobility. These resources are as follows: [35]

- *State* The state of an agent can be considered as a snapshot of its execution. It allows the agent to resume his execution as soon as he arrives at his destination.
- *Implementation* The mobile agent needs a code to run.
- *Interface* An agent provides an interface that allows other agents and other systems to interact with it.
- *ID* Each agent has a unique ID during its life cycle, which allows it to be identified and located.
- *Authority* An authority is an entity whose identity can be authenticated by any system it tries to access. There are two main types of authority, the manufacturer who is

the provider of the agents' implementation code and the owner who is responsible for the agents' behavior.

2.4 Operation of a Mobile Agent

The client gives a mission to an agent. To achieve this mission, the agent moves in the network of machines accessing the services offered by these machines locally. Three phases can be distinguished [4] as the Fig. 4 shows:

1. Activation of the mobile agent with the description of its mission.
2. The execution of the mission by the agent.
3. Possible recovery of the results of the mobile agent.

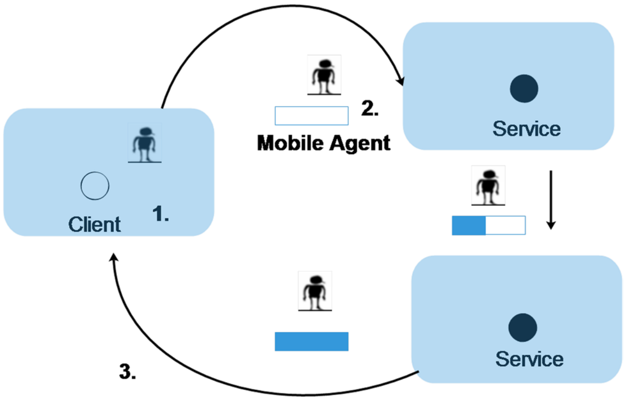


Fig. 4 Operation of a mobile agent

3 Mobile Agent Itinerary Planning

3.1 Static Itinerary Planning

Figure 5 illustrates the classification of different approaches based on the Itinerary planning algorithms.

3.1.1 Single Itinerary Planning

Local closest first (LCF) and global closest first (GCF) are the two first heuristics proposed algorithms in [37] to optimize the itinerary of MA performing data fusion tasks. In one hand, LCF plan the itinerary by starting from the sink then searches for the closest SN to its current location as the next destination, the same process is repeated until all SNs are included in the itinerary. Figure 6 illustrates LCF Algorithm . In the other hand, GCF searches for the SN with the shortest distance to the sink as its next destination SN instead of closest SN to its current location as in LCF. Figure 7 illustrates GCF Algorithm.

The output of LCF-like algorithms highly depends on the MA original location, while the SNs left to be visited last are

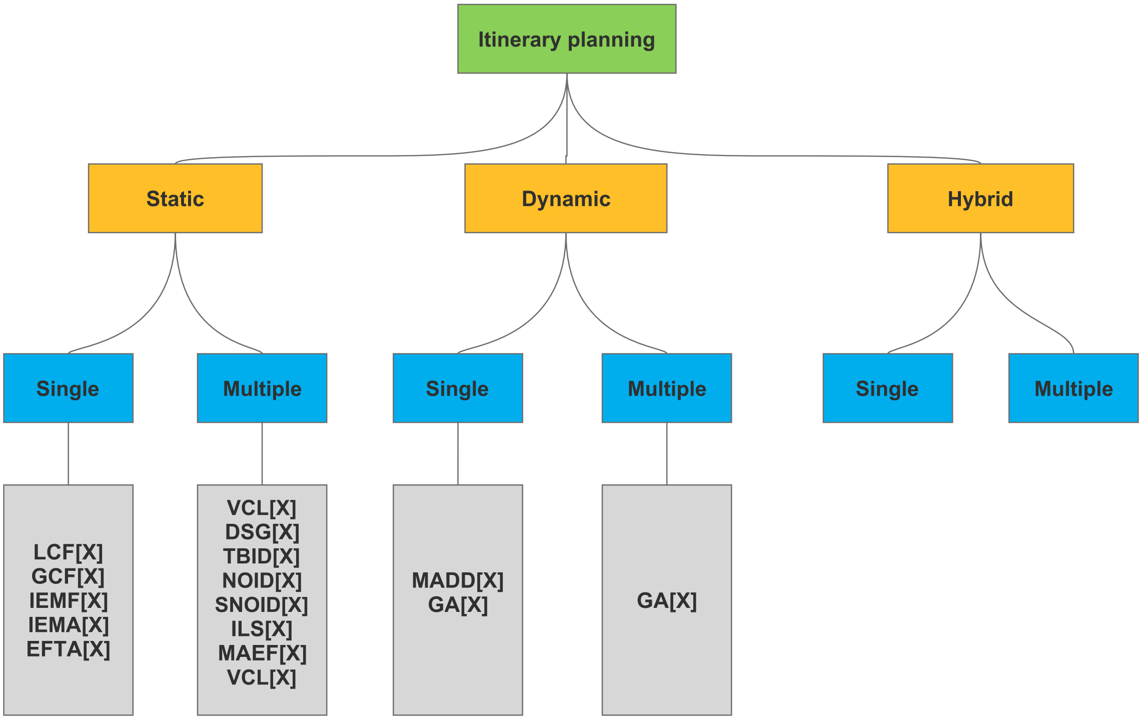


Fig. 5 Differents proposed approaches for itinerary planning for MAs in WSNs

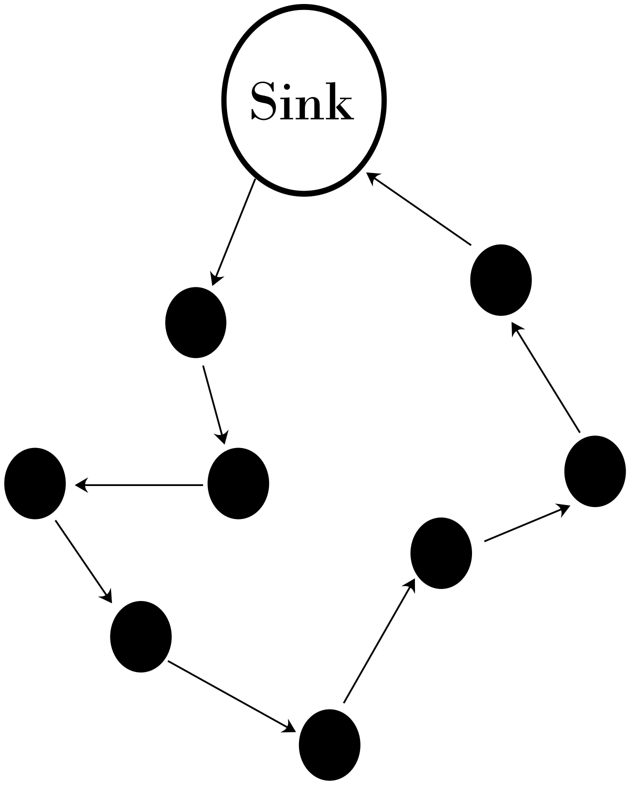


Fig. 6 Local closest first LCF

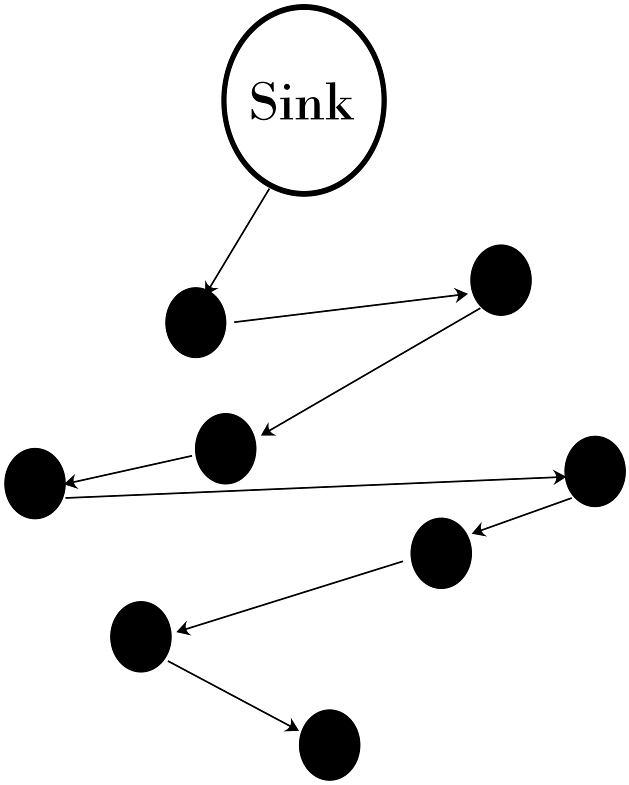


Fig. 7 Global closest first GCF

typically associated with high migration cost; the reason for this is that they searches for the next destination among the SNs adjacent to the MAs' current location, instead of looking at the global network distance matrix. On the other hand, GCF produces messier routes than LCF and repetitive MA oscillations around the region center, resulting in long route paths and unacceptably poor performance [37].

In [11] the authors propose a better scheme named IEMF itinerary energy minimum for first-source-selection (IEMF) algorithm, as well as the itinerary energy minimum algorithm (IEMA), the iterative version of IEMF. IMEF denotes the importance of choosing the first visiting node. Based on this conclusion, it estimates energy costs of different choices of the first node and adopts the best solution to achieve energy efficiency. During each iteration, IEMA selects an optimal source node as the next source to visit among the remaining set of source nodes.

Usman et al. [45] proposed a Genetic Algorithm (GA) to exploit the global information of sensor detection signal levels and link power consumption. This algorithm provides superior performance than the LCF and GCF algorithms ,but it implies a time-expensive optimal itinerary calculation (genetic algorithms typically start their execution with a random solution 'vector' which is improved as the execution progresses), which is unacceptable for time-critical applications. The original LCF, GCF [37] and GA schemes [45] are all based on the following two assumptions: (1) a cluster-based network architecture, where all nodes (e.g., sink and source nodes) can communicate with each other in one hop; (2) high redundancy among the sensory data, which can be fused into a single data packet with a fixed size. This implies that a perfect aggregation model is used. These assumptions limit the scope of the existing schemes.

Authors in [15–17] have proposed an approach based on SNs density. At first, SNs are grouped in clusters based on SNs density, then a SN is selected as the cluster head. second, minimum spanning tree is used to plan th itinerary amongst cluster heads. finally, MA is dispatched to collect data from cluster heads. In addition a fault tolerant is proposed in [17] in case of SN failure.

3.1.2 Multiple Itinerary Planning

Mpitziopoulos et al. [31] have proposed a near-optimal itinerary design (NOID) algorithm.This algorithm adapts a method originally designed for network design problems, namely the Esau-Williams heuristic [11] for the Constrained Minimum Spanning Tree (CMST) problem, in the specific requirements of WSNs. In NOID, the parallel deployment of multiple agents is suggested where each agent visits a subset of nodes. NOID suffers from low working speed and high computational complexity.

Chen [6] proposed scheme entitled MST-MIP based on minimum spanning tree, where each branch stemmed from the sink corresponds to a group of source nodes assigned for a mobile agent to visit. A SIP algorithm is then executed to decide on the order in which cluster nodes will be visited. Furthermore, a balancing factor a is introduced to achieve a flexible trade-off control between energy cost and task duration.

In [9] the Authors propose The Directional Source Grouping (DSG) algorithm to find near optimal itineraries for multiple agents. This Algorithm uses a circle around the PE and iteratively partitions a directional sector zone where the source nodes are included in an itinerary. The radius of the circle is set to the maximum transmission range of a single SN. Every SN that lies in the circle is used as the first node of an itinerary. The number of source nodes included in the itinerary is controlled by the angle of the directional sector zone in such a way that routes for MAs can be obtained by selecting the appropriate angle.

In [8], the authors proposed multi-agent itinerary planning (MIP) algorithms. The basic idea of the proposed visiting central location (VCL) selection algorithm is to distribute each source's impact factor to other source nodes to determine a set of SNs that will act as central points in each cluster. Then the itinerary for each of the Mobile agents can be planned by any SIP algorithms.

The Tree-Based Itinerary Design (TBID) algorithm [26] find near optimal itineraries for multiple agents. , the algorithm determines a spanning forest of trees in the network, calculates efficient tree traversal orders (itineraries), and eventually, assigns these itineraries to individual MAs. The main theme of the TBID algorithm is to divide the area around the sink into concentric zones to construct the near optimal itinerary tree from inner zones to the outer zones. . The first zone includes all SNs that are within a certain distance of the PE and a MA is assigned to each itinerary rooted at those SNs. They used post order traversal with a possible shortcutting of the itinerary tree to derive an itinerary for each agent. authors in Gupta et al. [27] have present a Scalable and Load-balanced Mobile Agents-based Data Aggregation (SLMADA) algorithm which is based on TBID with scalability and load-balancing.

Gavalas et al. [24] have proposed a meta-heuristic approach-based itinerary planning algorithm for multiple agents. Proposed method uses iterated local search (ILS) to improve the migration path of the MAs. Like TBID [26] , ILS also divides the area around the sink into a set of concentric zones and uses ILS algorithm for grouping the nodes in the separate itineraries. This method is generally formed radial-like itineraries that gradually grow up towards the border of the network. ILS scheme is centralized and does not deal with the MA bloated state problem.

In [23], the authors have proposed an itinerary algorithm for the MA, based on iterated local search approach. The proposed algorithm is designed for heterogeneous WSN with multiple mobile sinks. The main limitation of this algorithm is that it also derives static itineraries which may be based on the old observations of the network topology. Due to this reason, agents are unable to complete its itinerary when any nodes of the itinerary list become faulty or dead.

In [16, 18] authors have proposed a multiple itinerary planing approach based on SNs density. At first, a density impact factor is used to calculate the density in the network, then based on this density, SNs are grouped in clusters b,after that a SN is selected as the cluster head in each Cluster. In the next phase, minimum spanning tree is used to plan th itinerary amongst cluster heads. finally, MAs are dispatched to collect data from cluster heads. In addition a fault tolerant is proposed in [18] in case of SN failure.

Qadori et al. [36] have proposed a spawn multi-mobile agent itinerary planning scheme which is based on agent spawning approach. The proposed scheme first uses x-means clustering algorithm for the partitioning the network area into a set of clusters. This scheme uses LCF algorithm for deriving the itineraries of the agent in each partition based on the global topology information available at the sink. This scheme is also suffers from the same problem as scheme proposed in [23].

3.2 Dynamic Itinerary Planning

3.2.1 Single Itinerary Planning

A Single Dynamic Itinerary Planning Algorithm (SDIP) has been proposed in [10], the authors proposed the mobile agent-based directed diffusion (MADD). In this approach, authors adapt the directed diffusion [29] in CS paradigm to MA paradigm. Although quite similar to LCF, MADD selects the farthest SN from the sink as the first SN in the itinerary. Figure 8 illustrate MAAD algorithms.

[47, 49] proposed a Single dynamic agent migration algorithm for a target tracking application. In this Algorithm, a MA migrates to a source node that can get more accurate information about the target location by consuming less energy. to select the next node, they defined a cost function, which includes the following three components: energy consumption, information gain, and the remaining energy of a node. Once an agent accumulates sufficient information so that the accuracy of the estimation meets the desired level, the agent will terminate the migration and return to the sink [47, 48]. This algorithm is more time expensive and may face difficulty in returning back to the sink without additional forwarding information.

Qi and Wang [39] proposed a software agent-based directed diffusion where the order of node visits is

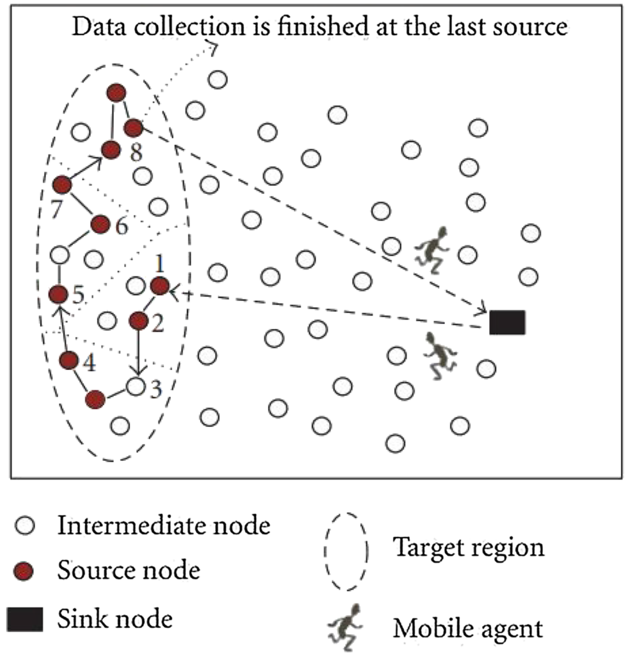


Fig. 8 Mobile Agent based Directed Diffusion (MADD)

determined at the sink node, but the authors did not describe the procedure. This method takes the routing cost and the remaining energy of a node for selecting a next node to be visited by an agent.

3.2.2 Multiple Itinerary Planning

Gupta et al. [25] proposed a multiple mobile agents with dynamic itineraries-based data dissemination (MMA-DIDD) protocol where nodes are organized in a set of the wedge regions and each agent is responsible for gathering aggregated data from each wedge region. The route of an agent is dynamically decided at each hop using cost function. MMADIDD adapts to unexpected node failures during an agent migration, but consumes slightly more energy than TBID [26]. This protocol provides a fair amount of fault tolerance, but the migration of agents depends on the wedge structure.

3.3 Hybrid Itinerary Planning

In hybrid planning, the selection of the source-visiting set is static, whereas the decision of the source-visiting sequence is dynamic. We did check the literature related to Hybrid Itinerary planning, No hybrid itinerary planning approach have bee, proposed.

4 Results and Discussion

We implemented the most prominent single itinerary planning and multiple itinerary planning algorithms. Taking into consideration that MIP outperforms SIP, we did compare SIP algorithms with each other and MIP with each other using castalia simulator [34]. Castalia is a simulator for WSNs, body area networks BANs and generally networks of low-power embedded devices. It is based on the OMNeT++ platform and used by researchers and developers to test protocols in realistic wireless channel and radio models, with a realistic node behavior especially relating to the access of the radio. Castalia can also be used to evaluate different platform characteristics for specific applications, since it is highly parametric, and can simulate a wide range of platforms.

SNs were randomly deployed in square monitoring area of 500 × 500 m, and varied from 200 to 800 nodes, the sink is located at the center of monitoring area, also SNs had the same transmission range and battery power, except for the sink, that has more computation capabilities and battery power.

The rest of the simulation parameters are shown in Table 2.

For data aggregation model used in this work, we adopted the model proposed in [11] :

$$S_{MA}^{k} = S_{MA}^{0} + S_{D} + \sum_{n=1}^{k-1}(1-\alpha) * S_{D} \tag{1}$$

Where S_{MA}^{k} is the size of MA after visiting *kth* SN, S_{MA}^{0} is the MA size after being dispatched by the sink for the first time, S_D is the size of data at each cluster head, and α is the data aggregation ratio. When data aggregation ration $\alpha = 1$:

$$S_{MA}^{k} = S_{MA}^{0} + S_{D} \tag{2}$$

In the other hand, where $\alpha = 0$ which means that there is no data aggregation taking place by mobile agents at cluster heads:

$$S_{MA}^{k} = S_{MA}^{0} + k.S_{D} \tag{3}$$

Table 2 Simulation parameters

Parameter	Value
Number of nodes	[200, 800]
Network transfer rate	250 Kbps
Monitoring field size	500 × 500
Node energy	18720 J
Energy consumed by MA execution	5 nJ
MA processing delay	50 ms
MA instantiation delay	10 ms
Collected data size at each node	200 Bytes
MA code size	1024 Bytes

The following four metrics were used to compare SIP and MIP:

- *Overall energy consumption* is the energy consumed by all SNs and MAs execution.
- *Execution time* is the required time for MAs to visit all CHs and returning back to the sink.
- *Total traveled distance* is the total traveled distance by all MAs.
- *Dispatched MAs* is the number of dispatched MAs to gather data from CHs (this metric is used for MIP algorithms).

Figure 9 illustrate the comparison of overall energy consumption per round of GCF, IEMF, MADD, EFTA and A-EFTA algorithms with the increasing size of network from 200 to 800 SN. As shown in Fig. 6, the overall energy consumption per round increases as the number of SNs increase. IEMF and MADD protocols consume almost the same amount of the energy, with better performance of IEMF. GCF consumes the biggest amount of energy thean all other algorithms. For MAEF it consumes almost the smallest amount of energy.

Figure 10 shows the execution time of MAs to visit all SNs including the time to return back to sink. It compares MAEF, GCF, MADD, EFTA and A-EFT Aalgorithms. As it can be observed, the execution time of all algorithmes increases as the number of SNs increases. GCF has the highest execution time than all other protocols, because of its poor strategy to migrate amongst SNs neare preforms better since it spends less time for visiting CHs only.

Figure 11 compares the total itinerary length of MAEF, IEMF, MADD EFTA and A-EFTA algorithmes. MAEF has the shortest itinerary length than all the other algorithmes. This is due to the itinerary planning just among CHs and not all the SNs. GCF has the longest itinerary length because its poor strategy of choosing nearest SN as the next destination and planning itinerary among all SNs. On the other side, MADD and IEMF have almost the same itinerary length.

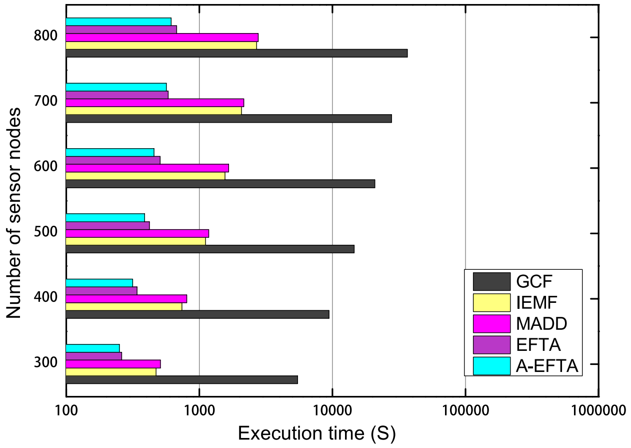


Fig. 10 Execution time

Figure 12, illustrate the comparison of overall energy consumption per round of TBID, VCL, DSG, NOID and MAEF algorithms with the increasing size of network from 200 to 800 SN. As shown in Fig. 12, the overall energy consumption per round increases as the number of SNs increase. NOID and VCL algorithms consume the highest amount of energy, with better performance of VCL followed by DSG algorithme. TBID and MAEF manages to consume less energy in comparaison to the all other algorithmes withe better performance of TBID when the network size increase.

Figure 13 shows the execution time required by the MAs to visit all SNs in the network including the time to return back to sink. It compares the execution time(task duration) of TBID, VCL, DSG, NOID and MAEF algorithms. As the

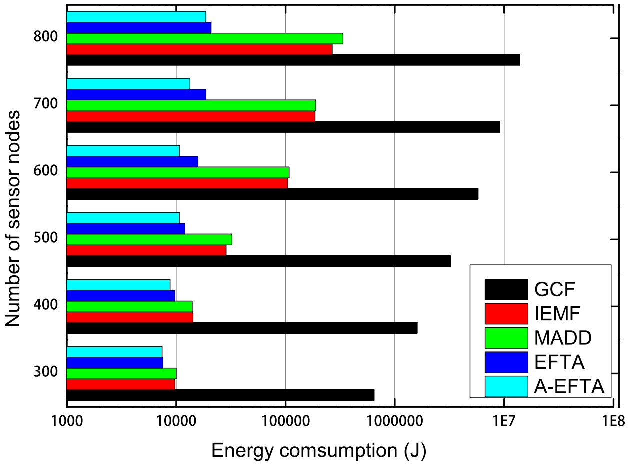


Fig. 9 Energy Consumption

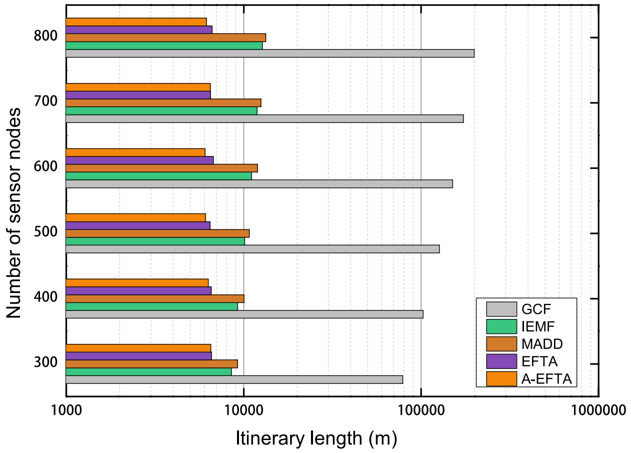


Fig. 11 Itinerary length

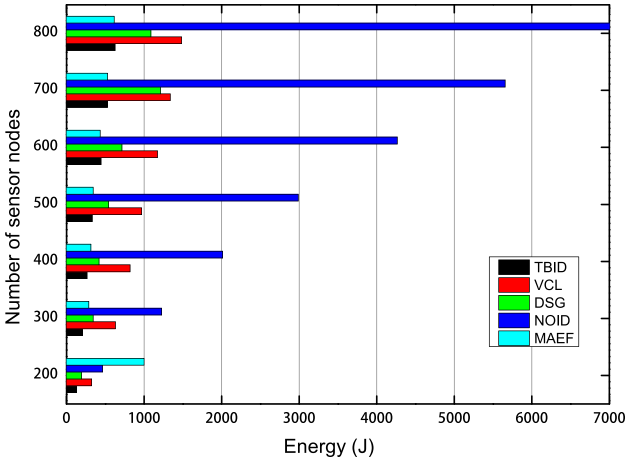


Fig. 12 Energy Consumption

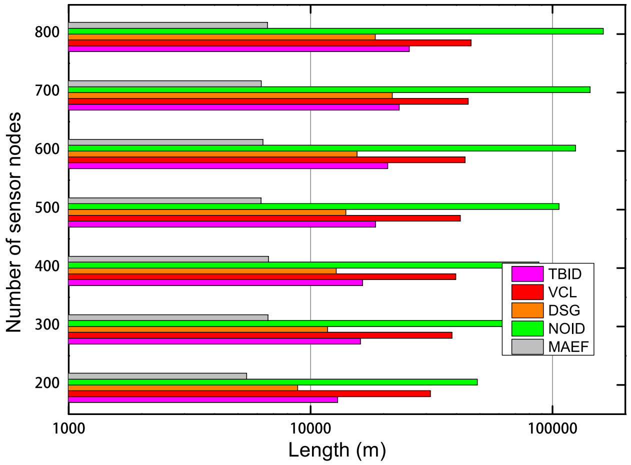


Fig. 14 Itinerary length

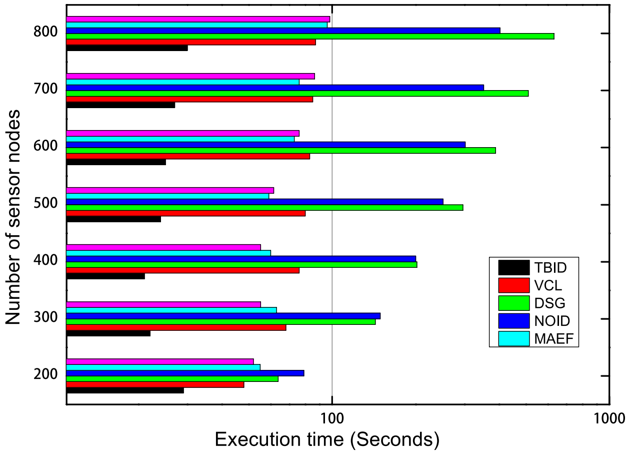


Fig. 13 Execution time

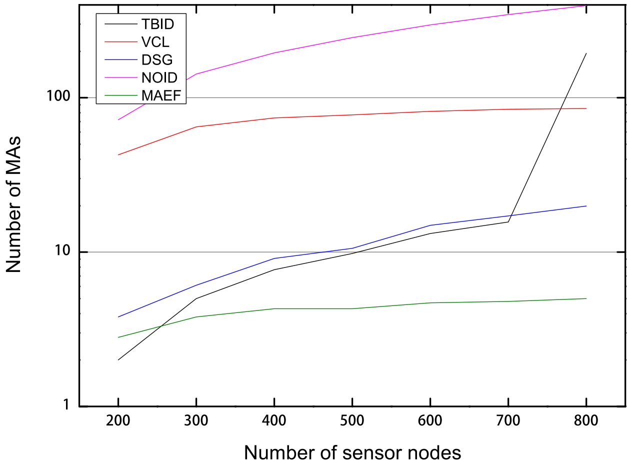


Fig. 15 Dispatched MAs

figure shows, the execution time of all algorithmes increases as the number of SNs increases. NOID and DSGhas the highest execution time than all other Protocols, followeb by MAEF and and VCL which have less execution time. TBID manages to consume less energy the all other protocoles.

Figure 14 compares the total itinerary length of TBID, VCL, DSG, NOID and MAEF algorithmes. NOID has the longest itinerary length than all other algorithms because it's a tree based algorithme. followed by VCL then TBID NOID and DSG. MAEF manages to have the shortest itinerary of all other algorithms. Because of its strategy of dispatching the minumuim number of MAs.

Finally, Fig. 15 shows the number of dispatched MAs by each algorithme. as the figure shows, NOID algorithm dispatches the highest number of MAs followed by VCL algrorithme. DSG and TBID manages to dispatche 20 MA in a network of 800 SN. MAEF dispatches the minimeum number of MAs (Fig. 15).

5 Summary and Perspectives

The mobile agents paradigm is currently adopted by many researchers in WSNs especially for data fusion application. In comparison to the traditional CS paradigm, MAs is efficient in terms of energy consumption, execution time and network lifetime. The efficiency of MA paradigm is largely related to the itinerary followed by the MA during the data fusion task. Thus, many approaches have been proposed to solve the problem of itinerary planning.

In this paper, we survey the literature related to the itinerary planning for MA in WSNs for data fusion application. We did simulate the most prominent approaches to evaluate and compare their performances. It is obvious that planning multiple itineraries for multiple agents is more complicated then planning single itinerary for one agents and it's a NP-hard problem as we mentioned in the manuscript. The simulation results shows that MIP outperform SIP.

Compliance with Ethical Standards

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Akkaya K, Younis M (2005) A survey on routing protocols for wireless sensor networks. Ad Hoc Netw 3(3):325–349. https://doi.org/10.1016/j.adhoc.2003.09.010
2. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks: a survey. Comput Netw 38(4):393–422. https://doi.org/10.1016/S1389-1286(01)00302-4
3. Alonso RS, García O, Zato C, Gil O, De la Prieta F (2010) Intelligent agents and wireless sensor networks: a healthcare telemonitoring system. In: Demazeau Y, Dignum F, Corchado JM, Bajo J, Corchuelo R, Corchado E, Fernández-Riverola F, Julián VJ, Pawlewski P, Campbell A (eds) Trends in practical applications of agents and multiagent systems. Springer, Berlin, pp 429–436
4. Brustoloni JC (1999) Autonomous agents: characterization and requirements
5. Cabri G, Leonardi L, Zambonelli F (2018) Weak and strong mobility in mobile agent applications. In: Proceedings of the 2nd international conference and exhibition on the practical application of Java (PA JAVA 2000), Manchester (UK)
6. Chen M, Cai W, Gonzalez S, Leung VCM (2010) Balanced Itinerary planning for multiple mobile agents in wireless sensor networks. In: Zheng J, Simplot-Ryl D, Leung VCM (eds) Ad Hoc networks. ADHOCNETS 2010. Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering, vol 49. Springer, Berlin
7. Chen M, Gonzalez S, Leung VCM (2007) Applications and design issues for mobile agents in wireless sensor networks. IEEE Wireless Commun 14(6):20–26. https://doi.org/10.1109/MWC.2007.4407223
8. Chen M, Gonzalez S, Zhang Y, Leung VCM (2009) Multi-agent Itinerary planning for wireless sensor networks. Springer, Berlin, pp 584–597. https://doi.org/10.1007/978-3-642-10625-5_37
9. Chen M, Gonzalez-Valenzuela S, Leung VCM (2010) Directional source grouping for multi-agent itinerary planning in wireless sensor networks. In: 2010 International conference on information and communication technology convergence (ICTC), pp 207–212. https://doi.org/10.1109/ICTC.2010.5674660
10. Chen M, Kwon T, Yuan Y, Choi Y, Leung VCM (2007) Mobile agent-based directed diffusion in wireless sensor networks. EURASIP J Appl Signal Process 2007(1):219–219. https://doi.org/10.1155/2007/36871
11. Chen M, Yang LT, K T, Zhou L, J M (2011) Itinerary planning for energ efficient agent communications in wireless sensor networks. IEEE Trans Veh Technol 3290–3299
12. Chess D, Harrison C, Kershenbaum A (1997) Mobile agents: are they a good idea? In: Vitek J, Tschudin C (eds) Mobile object systems towards the programmable internet. MOS 1996. Lecture notes in computer science, vol 1222. Springer, Berlin
13. Dhand G, Tyagi S (2016) Data aggregation techniques in wsn:survey. Procedia computer science. In: 2nd international conference on intelligent computing, communication and convergence, ICCC 2016, 24–25 January 2016, Bhubaneswar, Odisha, India 92:378–384. https://doi.org/10.1016/j.procs.2016.07.393
14. Dong M, Ota K, Yang LT, Chang S, Zhu H, Zhou Z (2014) Mobile agent-based energy-aware and user-centric data collection in wireless sensor networks. Comput Netw 74, Part B:58–70. https://doi.org/10.1016/j.comnet.2014.06.019 Special Issue on Mobile Computing for Content/Service-Oriented Networking Architecture
15. El Fissaoui M, Beni-Hssane A, Saadi M (2017) Energy aware hybrid scheme of client-server and mobile agent models for data aggregation in wireless sensor networks. In: Abraham A, Haqiq A, Alimi AM, Mezzour G, Rokbani N, Muda AK (eds) Proceedings of the 16th international conference on hybrid intelligent systems (HIS 2016). Springer, Cham, pp 227–232
16. El Fissaoui M, Beni-Hssane A, Saadi M (2017) Mobile agent protocol based energy aware data aggregation for wireless sensor networks. Procedia Computer Science 113:25–32. https://doi.org/10.1016/j.procs.2017.08.279. http://www.sciencedirect.com/science/article/pii/S1877050917316885. The 8th international conference on emerging ubiquitous systems and pervasive networks (EUSPN 2017)/The 7th international conference on current and future trends of information and communication technologies in healthcare (ICTH-2017)/affiliated workshops
17. El Fissaoui M, Beni-Hssane A, Saadi M (2018) Energy efficient and fault tolerant distributed algorithm for data aggregation in wireless sensor networks. J Ambient Intell Hum Comput. https://doi.org/10.1007/s12652-018-0704-8
18. El Fissaoui M, Beni-hssane A, Saadi M (2018) Multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks. EURASIP J Wireless Commun Netw 2018(1):92. https://doi.org/10.1186/s13638-018-1099-0
19. Ferrari G, Martalò M, Abrardo A (2014) Information fusion in wireless sensor networks with source correlation. Information Fusion 15:80–89. https://doi.org/10.1016/j.inffus.2012.09.001. http://www.sciencedirect.com/science/article/pii/S156625351200084X. Special Issue: Resource Constrained Networks
20. Fok C, Roman G, Lu C (2005) Mobile agent middleware for sensor networks: an application case study. In: IPSN 2005. Fourth international symposium on information processing in sensor networks, 2005, pp 382–387. https://doi.org/10.1109/IPSN.2005.1440953
21. Franklin S, Graesser A (1997) Is it an agent, or just a program?: a taxonomy for autonomous agents. In: In Intelligent Agents III. Springer
22. Fuggetta A, Picco GP, Vigna G (1998) Understanding code mobility. IEEE Trans Softw Eng 24(5):342–361. https://doi.org/10.1109/32.685258
23. Gavalas D, Venetis IE, Konstantopoulos C, Pantziou G (2017) Mobile agent itinerary planning for wsn data fusion: considering multiple sinks and heterogeneous networks. Int J Commun Syst 30(8):3184. https://doi.org/10.1002/dac.3184
24. Gavalas D, Venetis IE, Konstantopoulos C, Pantziou G (2016) Energy-efficient multiple itinerary planning for mobile agents-based data aggregation in wsns. Telecommun Syst 63(4):531–545. https://doi.org/10.1007/s11235-016-0140-z
25. Gupta GP, Misra M, Garg K (2012) Multiple mobile agents based data dissemination protocol for wireless sensor networks. In: Meghanathan N, Chaki N, Nagamalai D (eds) Advances in computer science and information technology. Networks and communications. Springer, Berlin, pp 334–345
26. Gupta GP, Misra M, Garg K (2014) Energy and trust aware mobile agent migration protocol for data aggregation in wireless sensor networks. J Netw Comput Appl 41:300–311. https://doi.org/10.1016/j.jnca.2014.01.003
27. Gupta GP, Misra M, Garg K (2017) Towards scalable and load-balanced mobile agents-based data aggregation for wireless

sensor networks. Comput Electr Eng 64:262–276. https://doi.org/10.1016/j.compeleceng.2017.10.020

28. Iabbassen D, Moussaoui S (2015) Data dissemination protocols in wireless sensor networks client/server versus mobile agent paradigms. In: Fifth international conference on the innovative computing technology (INTECH 2015), pp 45–50. https://doi.org/10.1109/INTECH.2015.7173477
29. Intanagonwiwat C, Govindan R, Estrin D, Heidemann J, Silva F (2003) Directed diffusion for wireless sensor networking. IEEE/ACM Trans Netw 11(1):2–16. https://doi.org/10.1109/TNET.2002.808417
30. Johansen D, van Renesse R, Schneider FB (1995) Operating system support for mobile agents. In: Proceedings 5th Workshop on Hot Topics in Operating Systems (HotOS-V), pp 42–45. https://doi.org/10.1109/HOTOS.1995.513452
31. Konstantopoulos C, Mpitziopoulos A, Gavalas D, Pantziou G (2010) Effective determination of mobile agent itineraries for data aggregation on sensor networks. IEEE Trans Knowl Data Eng 22(12):1679–1693
32. Krishnan D (2015) A distributed self-adaptive intrusion detection system for mobile ad-hoc networks using tamper evident mobile agents. Procedia Computer Science 46:1203–1208. https://doi.org/10.1016/j.procs.2015.01.034. http://www.sciencedirect.com/science/article/pii/S1877050915000356. Proceedings of the international conference on information and communication technologies, ICICT 2014, 3–5 December 2014 at Bolgatty Palace and Island Resort. Kochi, India
33. Lange DB, Oshima M (1999) Seven good reasons for mobile agents. Commun ACM 42(3):88–89. https://doi.org/10.1145/295685.298136
34. Pediaditakis D, Tselishchev Y, Boulis A (2010) Performance and scalability evaluation of the castalia wireless sensor network simulator. In: Proceedings of the 3rd international ICST conference on simulation tools and techniques, SIMUTools ’10, pp. 53:1–53:6. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium. https://doi.org/10.4108/ICST.SIMUTOOLS2010.8727
35. Peter Braun WRR (2005) Mobile agents: basic concepts, mobility models, and the Tracy toolkit
36. Qadori HQ, Zulkarnain ZA, Hanapi ZM, Subramaniam S (2017) A spawn mobile agent itinerary planning approach for energy-efficient data gathering in wireless sensor networks. Sensors 17(6):5. https://doi.org/10.3390/s17061280
37. Qi H, Wang F (2001) Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks
38. Qi H, Xu Y, Wang X, Member S (2003) Mobile-agent-based collaborative signal and information processing in sensor networks. In: Proceedings of the IEEE, pp 1172–1183
39. Shakshuki E, Malik H, Denko MK (2008) Software agent-based directed diffusion in wireless sensor network. Telecommun Syst 38(3):161–174. https://doi.org/10.1007/s11235-008-9102-4
40. Shalli Rani SHA (2016) Multi-hop routing in wireless sensor networks. Springer, Berlin
41. Sheltami T, Musaddiq M, Shakshuki E (2016) Data compression techniques in wireless sensor networks. Future Gener Comput Syst 64:151–162. https://doi.org/10.1016/j.future.2016.01.015
42. Tseng Y, Kuo S, Lee H, Huang C (2004) Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. Comput J 47(4):448–460. https://doi.org/10.1093/comjnl/47.4.448
43. Usman M, Muthukkumarasamy V, Wu XW, Khanum S (2012) Securing mobile agent based wireless sensor network applications on middleware. In: 2012 International symposium on communications and information technologies (ISCIT), pp 707–712. https://doi.org/10.1109/ISCIT.2012.6380993
44. Vasuhi S, Vaidehi V (2016) Target tracking using interactive multiple model for wireless sensor network. Inf Fusion 27:41–53. https://doi.org/10.1016/j.inffus.2015.05.004
45. Wu Q, Rao NSV, Barhen J, Iyengar SS, Vaishnavi VK, Qi H, Chakrabarty K, Member S, Member S (2004) On computing mobile agent routes for data fusion in distributed sensor networks. IEEE Trans Knowl Data Eng 16:740–753
46. Xu Y, Qi H (2004) Distributed computing paradigms for collaborative signal and information processing in sensor networks. J Parallel Distrib Comput 64(8):945–959. https://doi.org/10.1016/j.jpdc.2004.04.002
47. Xu Y, Qi H (2007) Dynamic mobile agent migration in wireless sensor networks. Int J Ad Hoc Ubiquitous Comput 2(1/2):73–82. https://doi.org/10.1504/IJAHUC.2007.011605
48. Xu Y, Qi H (2008) Mobile agent migration modeling and design for target tracking in wireless sensor networks. Ad Hoc Netw 6(1):1–16. https://doi.org/10.1016/j.adhoc.2006.07.004
49. Zhao F, Guibas LJ, Guibas L (2004) Wireless sensor networks: an information processing approach. Morgan Kaufmann Publishers Inc., San Francisco

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.