



# An energy efficient multi-mobile agent itinerary planning approach in wireless sensor networks

Tariq Alsboui<sup>1</sup> · Yongrui Qin<sup>1</sup> · Richard Hill<sup>1</sup> · Hussain Al-Aqrabi<sup>1</sup>

Received: 10 November 2020 / Accepted: 24 June 2021 / Published online: 14 July 2021

© The Author(s), under exclusive licence to Springer-Verlag GmbH Austria, part of Springer Nature 2021

## Abstract

Mobile Agent (MA) technology brings many benefits into Wireless Sensor Networks (WSNs), such as saving network bandwidth and enabling energy efficient mechanisms for collecting sensor data. Nowadays, itinerary planning for MAs is one of the most important features of the WSN. However, the way in which all dispatched MAs are routed inside the sensor networks must be intelligently planned to reduce energy consumption and improve information accuracy. There have been many research efforts designing itinerary planning algorithms to deploy multiple MAs in a given sensor network, where routes are generated so that MAs can follow different routes to collect data from sensor nodes efficiently and effectively. This paper proposes a new energy efficient Graph-based Static Multi-Mobile Agent Iterinary Planning approach (GSMIP). GSMIP applies Directed Acyclic Graph (DAG) related techniques and divide sensor nodes into different groups based on the routes defined by MAs itineraries. MAs follow the predefined routes and only collect data from the groups they are responsible for. The experimental findings demonstrate the effectiveness and superiority of the proposed approach compared to the existing approaches in terms of energy consumption and task delay (time).

**Keywords** Mobile agent MA · Wireless sensor networks WSN · Itinerary planning

**Mathematics Subject Classification** 68W15

---

✉ Tariq Alsboui  
tariq.alsboui@hud.ac.uk

Yongrui Qin  
y.qin2@hud.ac.uk

Richard Hill  
r.hill@hud.ac.uk

Hussain Al-Aqrabi  
h.al-aqrabi@hud.ac.uk

<sup>1</sup> School of Computing and Engineering, University of Huddersfield, Huddersfield, UK

## 1 Introduction

A pervasive interconnected network, including a wireless sensor network (WSN), is defined by its capacity to perform basic tasks through exchanging resources that are in network or in node domains.

One primary aim of WSNs is to allow users to access information of interest from data obtained through spatially distributed sensors. Sensors are generally installed in large numbers to gain full visibility of the controlled physical environment. Such sensor network systems are designed in a way that immense amounts of data will be produced [1]. Mobile agent techniques have been widely used to enable efficient collaborative data collection from a WSN. In these techniques, mobile agents (MAs) will be dispatched, which will traverse the sensor along predefined routes, generated by itinerary planning, to collect data from sensor nodes on the way. The need to locate and handle mobile agents in energy-efficient WSN applications is primarily characterised by approaches used to design the itinerary planning of MAs.

Practical constraints on the implementation of sensor nodes, such as computational capacity and battery-limited powered make itinerary planning a challenging [2]. The critical issues while dispatching a mobile agent include the migration cost of mobile agent, itinerary planning and the approaches to establishing such a plan.

The principal objective of the mobile agent is to collect and process data in a network. Without user interaction, they can combine and make local decisions autonomously. The main reason why mobile agents are used is that radio communication is one of the most effective hungry operations [3]. To avoid long distance radio communication, we therefore dispatch agents to gather data instead of sending it back to a sink node. In such scenarios, planning mobile agent itinerary in order to optimize energy consumption for sensor nodes is critical. However, it has been challenging to solve the problem, which is NP-hard, of finding an ideal sequence of sensor nodes to be visited by a mobile agent [4,5].

Hence, one main challenge is how to create an appropriate itinerary for MAs to collect data [2]. Itinerary planning refers to identifying a route of a MA, which the MA should follow when traversing the sensor network and visiting sensor nodes. Each route contains a sequence of source nodes to be visited through the MA migration trip. Current techniques for the development of MA itineraries can be generally classified into three types: Static itinerary, Dynamic itinerary, and Hybrid itinerary [2], which will be discussed in more detail below.

### 1.1 Components of mobile agent

In WSNs, Mobile Agents (MAs) are referred to as a software abstractions performing information-rich data collection and autonomous data processing whilst dynamically migrating between network nodes so that data is exchanged between participant nodes [6].

MAs have also recently been suggested to address the limitations of centralized models' scalability and the flexibility problems of static hierarchical frameworks. MAs comprise of four components as shown in Fig. 1.

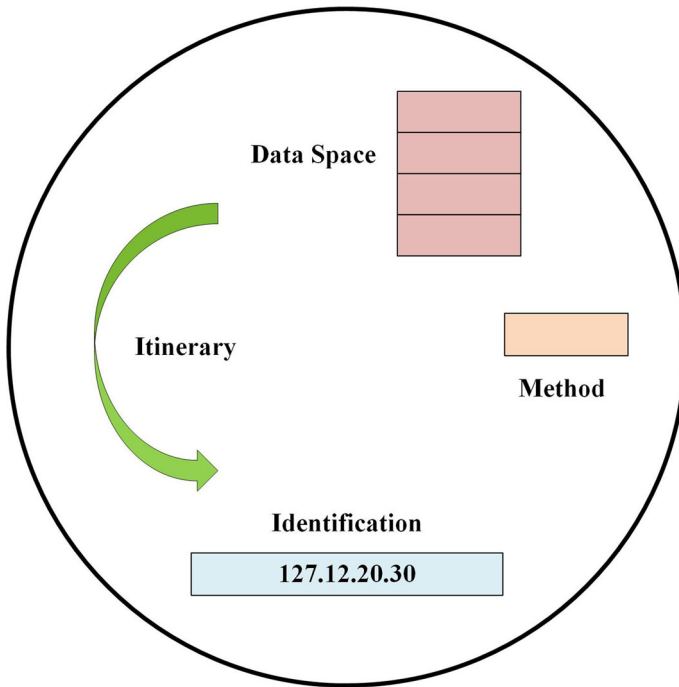


Fig. 1 Components of mobile agent

- *Itinerary* It can be identified as the mobile agent trip route for visiting source nodes. Itinerary planning is usually divided into three categories: static, dynamic and hybrid. In a static itinerary, the route is computed at the dispatcher prior to the MA migration. In a dynamic itinerary, the route of MA is determined by the MA on the fly. In a hybrid itinerary, the sensor nodes to be visited by the MA are selected by the dispatcher, but the visiting sequence is determined by the MA on the fly.
- *Data space* This is the data buffer of MA, and is primarily capable of producing data integration. The findings should have incremental precision as the agent moves from one node to another.
- *Identification* This is a unique number that identifies the mobile agent and the dispatcher. Typically presented in a 2-tuple (i:j) format, where i denotes the dispatcher's IP address, and j is a serial number assigned to each MA by the dispatcher.
- *Method* This is the execution code that each MA executes.

**Contribution:** Our work focuses on developing a Graph-based Static Multi-Mobile Agent Iterinary Planning approach (GSMIP), which applies Directed Acyclic Graph (DAG) in planning and dispatching mobile agents.

Our main contributions can be summarized as follows.

- We propose a new approach for multi-mobile-agent itinerary planning that manages resources with better energy efficiency and scalability.

- We propose a novel approach called GSMIP. GSMIP compute mobile agent itinerary planning based on Directed Acyclic graph (DAG) modelling. It divides DAG into groups and allocating mobile agents to different groups thereby achieving energy efficiency.
- Finally, we carry out a number of experiments to verify the efficiency and effectiveness of the proposed approach.

The remaining structure of this paper is organized as follows: Sect. 2 describes the related work and places our work in conjunction with the existing work. Section 3 details our proposed approach. In Sect. 4, we discuss the benefits of the proposed solution and compares it with alternative approaches. Lastly, in Sect. 5, we conclude the paper and discuss future work.

## 2 Related work

### 2.1 Single itinerary planning

Over the last few years, mobile agent itinerary planning has drawn many researchers' attention in the field of WSNs. We refer the interested readers to the recent survey [7] and the references therein for a comprehensive review of the mobile agent itinerary planning approaches in WSNs. It is noticed that many of these research efforts are towards optimizing and constructing an energy efficient itinerary planning mechanism [8–15]. One piece of the early work on Single Itinerary Planning (SIP) is proposed in [8], in which the authors have developed two heuristic algorithms to calculate the itinerary of the single mobile agent. Two algorithms are named local closest first (LCF) and global closest first (GCF) are proposed. LCF operates by finding the next node in the shortest distance to the current node while GCF aims to find the centre's closest node. The proposed algorithms are static and can save energy as the itinerary planning needs calculated only once.

However, the approach does not scale well if a single MA has to visit thousands or millions of sensor nodes. It also leads to big delays in reporting the data because of using only a single MA, which has to move between all sensor nodes in the network.

In [4] an event-driven adaptive method is proposed, which implements a semi-dynamic routing strategy based on a two-level genetic algorithm. A fitness function is constructed to meet the desired detection accuracy while minimizing energy consumption and path losses in a global sense. The sink node has necessary predetermined knowledge for performing the global optimization, such as the geographical locations of all sensor nodes. The sink node is responsible for computing the routes for mobile agent. The mobile agent follows the route computed by the genetic algorithm according to the fitness function. The proposed approach is energy-efficient. However, it lacks scalability since a single mobile agent is used, which is not suitable for time critical applications that require real time processing.

The authors in [12] proposed a mobile agent directed diffusion (MADD) approach, which is based on a directed diffusion algorithm. The approach works by making the sink to initially get diffused with an interest for notifications of low-rate exploratory

events that are intended for path setup and repair. The proposed approach reduces energy consumption because it relies on directed diffusion agent trip and eliminates data redundancy. However, it introduces a delay since a single mobile agent is routed among sensor nodes and is not suitable for large scale sensor networks.

The mechanism introduced in [13] is an improvement over the MADD approach. There are three phases of the proposed approach: First, the MA action phase; Second, the dissemination phase of exploratory data; And third, the controlled setup phase of gradients. In the first phase, the sink node floods its neighbor with interest messages in the controlled setup phase of gradients. It sets up an itinerary to the next hop based on two metrics: (1) the remaining energy threshold and (2) the minimum hop count. In the exploratory data dissemination phase, the main aim is to discover the source nodes and to establish the TargetSrcTable (containing targets and source nodes information) in each target node. Sensory data is stored in the cache of each source node, waiting for collection in the next phase. The approach is energy efficient. However, due the same limitations like above research efforts, the solution lacks scalability due to the use of a single MA.

The work proposed in [16] is called Itinerary Energy Minimum Algorithm (IEMA). IEMA extends LCF by considering estimated communication costs. The aim of IEMA is to achieve better energy-efficiency. It focuses on choosing the first visiting node among the remaining set of source nodes as well as an optimal source node as the next source node to be visited. The algorithm estimates the energy costs of the alternative choices of the first node. The proposed schema is energy-efficient. However, it does not scale to a large number of sensor nodes since only a single mobile agent is used. In addition, it does not take into consideration the growing size of collected data of the mobile agent when visiting a sequence of nodes.

## 2.2 Multiple itinerary planning (MIP)

To alleviate the inherent problem caused by the use of a single mobile agent, a number of Multiple Itinerary Planning (MIP) approaches have been developed [17].

Authors in [9] investigate the role of multiple mobile agent and propose a novel a routing itinerary algorithm called DMAIP. The idea is to group all sensor nodes into multiple itineraries for mobile agent. The approach consists of three main components, including: remote user, sink node and sensor node. The remote user assigns task to a sink node. When the sink node receives a task it traverses network topology to generate a spanning tree, and assigns each path to one of the mobile agents.

The authors in [18] propose directional source grouping algorithms (DSG-MIP). The main idea is to divide the network area into sector zones with specific angles, the centres of which are the immediate neighbors of the sink node. After this, the source nodes are allocated to an itinerary within each sector zone. The route to the sink node inevitably converges on each MA's round trip and increasingly extends as the MAs travels further from the sink node [10].

A new immune inspired algorithm, called the Clonal Selection Algorithm for Multi-agent Itinerary Planning (CSA-MIP), is proposed by the authors in [11], in order to solve the MIP problem in WSNs. The important components of CSA-MIP include

encoding, mutation operators, cloning of antibodies, and affinity function. CSA-MIP is based on two computational stages called Stage I and Stage II. Each stage involves a different mutation operator. The proposed approach has less computational complexity and is energy efficient.

A novel central location-based MIP (CL-MIP) framework is presented in [19]. The framework consists of four parts including: visiting central location (VCL) selection algorithm, source-grouping algorithm, SIP algorithm and its iterative algorithm. The VCL selection algorithm is used to calculate a high source node density. The source-grouping algorithm is responsible for grouping nodes and assign mobile agent to a particular group. The SIP algorithm is adopted to determine the itinerary of mobile agents. Finally, the iterative algorithm is mainly concerned with ensuring that all source nodes are assigned to the allocated MAs. The CL-MIP algorithm considers a cluster-based technique in which the source nodes are arranged geographically and distributed in several clusters. This indicates that the CL-MIP is not applicable to be used if the nodes are sparsely deployed.

In [14], an energy efficient itinerary planning approach is proposed. The algorithm is based on Iterated Local Search (ILS), a metaheuristic method commonly used for solving discrete optimization problems. ILS iteratively applies a simple modification to a local search routine, each time starting from a different initial configuration, in search of an improved solution. The ILS algorithm is executed centrally at the sink which statically determines the number of MAs that should be used and the itineraries these MAs should follow. The proposed algorithm is energy efficient and avoids delay. However, the itinerary planning is deterministic and pre-defined at the sink node. Therefore, if a sensor node depletes in energy, it would result in re-constructing the paths for each mobile agent.

The authors in [15] proposed a system called MAMS, which employs both mobile agent and mobile server to collect data from sensor nodes deployed in a sensing field. Mobile agents migrate from node to node autonomously and return to the mobile server after data collection. The migration process relies on a geographic routing approach to route mobile agents. Upon collecting data mobile agents find the current location of the mobile server and return to it with the aggregated data. The system focuses on a effective and intelligent gathering mechanism.

A multi-mobile agent itinerary planning-based energy and fault aware data aggregation (MAEF) method is proposed in [20] to plan itineraries for MAs. The approach comprises of three main phases, including: (1) cluster head selection and construction, (2) cluster head based itinerary planning and (3) mobile agent migration and data collection. In the cluster head selection phase, the idea is to distribute the density impact factor of each node to other sensor node, then the sensor node with the highest accumulated impact factor will be selected as a cluster head node. In the cluster head itinerary planning phase, mobile agent itineraries among cluster head nodes are constructed based on a minimum spanning tree (MST). In the final phase, the sink dispatches mobile agents to gather data from cluster head nodes.

In [21], a multiple mobile agent itinerary planning approach named as GIGM-MIP is proposed, which works in three phases. In the first phase, the network is partitioned using the k-means method and based on geographical information in which a set of partition is generated, and each partition can get a several mobile agents. In the second

phase, the number of mobile agents is determined, and group of nodes are defined for each mobile agent. Finally, the third phase is concerned with defining the itinerary that passes throughout the source nodes grouping of each mobile agent. Several mobile agents can be allocated to each partition.

The authors in [22] proposed a Scalable and Load-balanced Mobile Agents-based Data Aggregation (SLMADA) protocol, in which the itinerary of a mobile agent is dynamically decided at each hop. The whole monitoring area is divided into centric zones and it is assumed that the sink node knows the location coordinates of source nodes. A zone coordinator is selected in each concentric zone, which assists the MA in the dispatching and receiving to and from the network. The sink node will create a set of MAs, one for each zone coordinator and dispatches mobile agents to the centric zones. This approach allows mobile agents to decide their visiting sequence dynamically based on information provided by the zone coordinators.

Energy-Aware Mobile Agent Based (EAMB) is proposed in [23]. The network is divided into multiple clusters and a group of sensor nodes are assigned to one cluster. The itineraries of the MAs among cluster head is defined by using a minimum spanning tree (MST).

In [24], a dynamic and distributed migration protocol, called energy and trust aware mobile agent migration (ETMAM), is proposed. The main idea is to identify and bypass the faulty or malicious nodes during mobile agent migration process. The sink node dispatches mobile agents concurrently to the coordinator nodes of each wedge region in the network. Coordinator nodes are the nodes of innermost concentric ring. Due to the need of detecting malicious nodes, the whole approach is considered complicated and requires heavy computation.

In [25], the authors propose a spawn multi-mobile agent itinerary planning (SMIP) to reduce significant rises in energy costs and time spent on data collection. This is based on the spawning agent, which allows the primary MA to spawn another MA into a single fraction. The proposal uses k-means algorithms to calculate the number of clusters based on Bayesian ratings. The sink node specifies the number of MA and their itineraries for each partition when the partitioning is complete. The SMIP approach is similar to the proposed approach in this paper. SMIP applies the k-means technique to partition the network where our approach uses Directed Acyclic Graph (DAG) to model the network, based on which efficient MA itineraries are planned and sensors nodes are divided into groups.

To summarise, most of the above approaches do not take into consideration that when the mobile agent moves along the routes, the size of collected data from sensor nodes increases rapidly, leading to higher consumption of network bandwidth.

The evaluation of the recent mobile agent itinerary planning literature covers 18 representative approaches. Table 1 presents the comparisons of existing research from a number of angles, including scalability, energy-efficiency, grouping strategy, and delay. From Table 1, we can see that all SIP approaches lack scalability, and suffer from delays in reporting data back to a sink node during mobile agent migration. Meanwhile, all MIP approaches can help address the issues of scalability, and avoid excessive delays.

**Table 1** Comparison among mobile agent (MA) approaches

Categories	Approach	Scalability	Energy-efficiency	Grouping	Delay
SIP	[8]	×	✓	×	✓
	[4]	×	✓	×	✓
	[12]	×	✓	×	✓
	[13]	×	✓	×	✓
	[16]	×	✓	×	✓
MIP	[9]	✓	✓	✓	×
	[10]	✓	✓	✓	×
	[11]	✓	✓	✓	×
	[14]	✓	✓	✓	×
	[19]	✓	✓	✓	×
	[15]	✓	✓	✓	×
	[20]	✓	✓	✓	×
	[18]	✓	✓	✓	×
	[21]	✓	✓	✓	×
	[22]	✓	✓	✓	×
	[23]	✓	✓	✓	×
	[24]	✓	✓	✓	×
	[25]	✓	✓	✓	×

### 3 Proposed GSMIP itinerary planning approach

#### 3.1 GSMIP architecture

Figure 2 presents an abstract view of the proposed Graph-based Static Multi-Mobile Agent Iterinary Planning approach (GSMIP). It describes all relevant components including mobile agents, itinerary, sensor nodes, and collected data each of which is responsible for a specific task. It shows a set of sensor nodes in each route and how nodes are grouped together. The grouping is based on the identified shared nodes. A shared node is considered as a node with multiple routes and further described in Fig. 4. The routes are generated to cover all nodes in the network. The sink node is responsible for dispatching MAs to a particular group in order to collect data from. The MAs collect data from the groups that they are assigned to. For example, the itinerary (e.g., orange lines) represents the routes to the assigned group.

To prevent delays in reporting data and to facilitate local interactions, we use multiple MAs. We apply a similar approach as defined in [26,27] for the calculation of the size of sensory data to be collected by MAs. We employ similar data aggregation methods to remove redundancy and inconsistency. The effects of sensory data are combined with an aggregation ratio ( $\rho$ ,  $0 \leq \rho \leq 1$ ).

$L_{ma}^i$  is defined as the amount of accumulated sensory data after an MA collects data from source  $i$ .  $A_i$  is the amount of sensory data to be aggregated by  $\rho$ , which is the fusion factor.



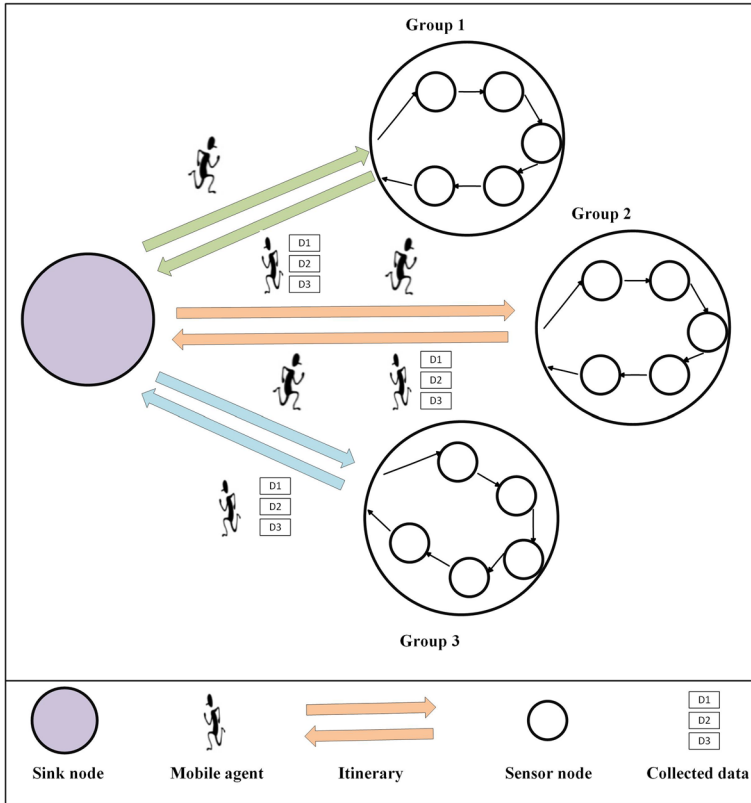


Fig. 2 The proposed mobile agent itinerary planning approach

$$L_{ma}^i = A_i$$

$$L_{ma}^2 = A_i + (1 - \rho) \times A_2 \tag{1}$$

$$L_{ma}^i = L_{ma}^i + (1 - \rho) \times A_2 \tag{2}$$

$$= A_1 + \sum_{g=2}^1 (1 - \rho) \times A_g \tag{3}$$

There is no data aggregation in the first node according to Eq. (3), and the value of  $\rho$  relies on the application type being deployed.

The proposed GSMIP data packet format is defined in Fig. 3. *FirstNode* represents the MA's first source node for data collection. *Static Routes* represent all the allocated nodes to be visited by the MAs. We have defined routes and groups. All computed routes are generated in which MAs will migrate to every node on the route whereas a group only determines the nodes that MAs should collect data from.

The payload of the agents is the pair of *Itinerary Planning* and *List of data*. The *Dispatcher ID* identifies the root node which dispatches and creates the current MA. *ToVisitFlag* is configured to determine whether or not the node was visited by an agent.

Consequently, routes are computed for MAs to migrate to every node and a group that defines the set of nodes to be visited i.e., collecting data from those nodes on this particular route.

In addition, to model a real-life situation, we believe that sensor devices are likely to produce highly similar data in close proximity. Accordingly, agents may delete redundant data through fusion.

### 3.2 Graph-based static multi-mobile agent itinerary planning algorithms

This part presents a description of the algorithms used in the proposed GSMIP approach.

The pseudocode to generate a random DAG  $G$  is presented in Algorithm 1, which is used for creating simulation scenarios. Algorithm 2 provides pseudocode for computing routes and generating groups of nodes. Algorithm 3 presents the pseudocode of multi-mobile agent dispatched to start sensory data collection.

At the outset, we use a random DAG as defined in (Algorithm 1), to construct a graph with a random number of edges and nodes. The iterated algorithm adds the necessary number of  $nodeNum$  nodes. After this it tests whether the  $G$  graph is a fully connected DAG, i.e., all source nodes are reachable from the sink node. Then, the algorithm applies the Depth-first search (DFS) technique to ensure that all nodes can be traversed back to the root.

---

**Algorithm 1:** Generate a random directed acyclic graph  $G$

---

**Input:**  $nodeNum, edgeNum$

**Output:**  $G$

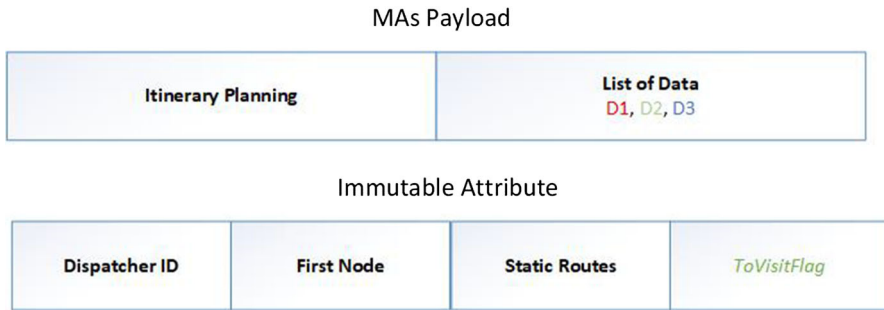
```

1 Initialize  $G$  to a directed acyclic graph (DAG) with  $nodeNum$  nodes but without
  any edges, and nodes range from 0 to  $nodeNum - 1$ 
2 while  $edgeNum \geq 0$  do
3    $node_a \leftarrow \text{randint}(0, nodeNum)$ 
4    $node_b \leftarrow node_a$ 
5   while  $node_b == node_a$  do
6      $node_b \leftarrow \text{randint}(0, nodeNum)$ 
7     Add  $\text{edge}(node_a, node_b)$  to  $G$ 
8     if  $G$  is still DAG then
9        $edgeNum \leftarrow edgeNum - 1$ 
10    else
11      Remove  $\text{edge}(node_a, node_b)$  from  $G$ 
12  Get DFS post-route ( $G$ )
13  Return DFS post-order
14 Return  $G$ 

```

---

In a typical WSN deployment, communication links between nodes can be either symmetric or asymmetric. In our case, we have assumed asymmetric communication links because that will make our approach more general. If communication links are



**Fig. 3** Message format of the proposed itinerary planning approach

symmetric our approach can still be applied by using only one of the communication links.

The Algorithm source grouping of sensor nodes (Algorithm 2) uses  $G$  Algorithm 1 as input and is designed specifically for making a group of source nodes for all mobile agents. It checks If the node has at least one connection or more in-degree connection and if the node has two or more connection out-degree connection. Hence, this is a shared node. It plays an important role within the network because it has multiple routes and is considered as a building block in generating groups. In addition, it acts as a main hub for connecting the nodes within the DAG network.

The algorithm finds a set of routes covering all nodes in the network by identifying the roots and the leaves. Then, it finds all possible routes between leaves and roots. For finding the least route, the source grouping algorithm sorts out all routes and compares them to identify the route with the least number of nodes. We define a shared node as the node that belongs to multiple routes. This means shared nodes can be reached by multiple routes. Shared nodes among different routes will be allocated as follows: (1) Initially each route is assigned a group of nodes, which belong to each particular route only (or we can call these nodes as private nodes on the route); (2) A shared node will be allocated to the group currently with the least number of nodes among the associated routes. Each generated group defines a set of nodes for the dispatched MAs to collect data from.

Figure 4 describes an example of the working principles of the algorithms. It shows a set of routes (route 1, route 2 and route 3 in this example) that cover all nodes, including private nodes and shared nodes, in the network. Private nodes are nodes that belong to a particular route only. Shared nodes are source nodes that are on multiple routes. There is only one shared node in this example, and it is on both route 1 and route 2. In addition, a group is a collection of private nodes and allocated shared nodes in a particular route. The groups are generated based on allocating shared nodes to the group of a route with the least number of nodes. Take Fig. 4 as an example. Since the group of nodes for route 2 has only two private nodes, while that of route 1 has three private nodes, we allocate the shared node to the group for route 2. Note that, the source sink (e.g., dispatcher) and the sink node (e.g., destination) in practice can be the same (sink) node. Here, because we model the whole network as a DAG, we

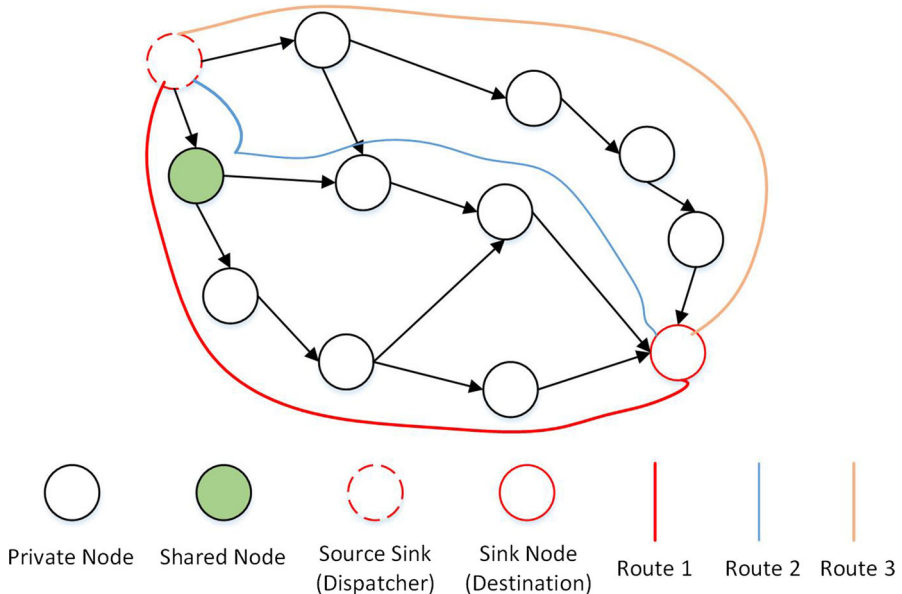


Fig. 4 An example of the working principles of the algorithms

virtually divide it into two nodes, where each node will make use part of the links to source nodes in the network.

---

**Algorithm 2:** Source grouping of sensor nodes

---

**Input:** DAG  $G$

**Output:** List of routes, Groups of source nodes

- 1 **if** *in-degree connection of node*  $\geq 1$  **and** *out-degree connection of node*  $\geq 2$  **then**
  - 2    | Shared Node in  $G$
  - 3 **for** *root in Roots in DAG* **do**
  - 4    | **for** *leaf in Leaves in DAG* **do**
  - 5    |    | **for** *route in nx.all simple paths (Dag, root, leaf)* **do**
  - 6    |    |    | Find all routes in  $G$
  - 7 **for** *each shared node routes list in DAG* **do**
  - 8    | **if** *route*  $\geq 2$  **then**
  - 9    |    | Select the route with the least number of nodes
  - 10 **for** *each selected least route* **do**
  - 11    | Add all least route nodes into a group.
  - 12 **Return** list of routes, Groups of source nodes
- 

Algorithm 3 dispatches mobile agents to groups. It begins with input (1) DAG  $G$ , List of routes  $R \in R$  and a groups of source nodes  $gs$ , given through Algorithm 2.

After this, the  $T$  will be initialised as a vacant bundle of  $MA$  data. It begins by dispatching mobile agents on a particular group in  $gs$ , guaranteeing that there are no two agents taking the same group. Every  $MA$  visits nodes on the given group  $gs$  during the trip. First, it verifies whether one of any mobile agent has visited the current node and decides whether to collect data from it. It directly proceeds to visit the next node on the route if the flag *visited* of the node is *true*. If not, the  $MA$  gathers data up towards its  $d$  data load and assigns the *visited* flag of this node as *true*. The  $MA$  finishes the tasks assigned and exits either by visiting all nodes on the specified route or by collecting  $d$  data on the trip. For each agent, the  $d$  data load threshold guarantees that in one trip the agent buffer is not overloaded with data.

---

**Algorithm 3:** Dispatch mobile agents  $MAs$  to collect data

---

**Input:** DAG  $G$ , list of routes  $R$ , Groups of source nodes  $gs$

**Output:** data  $T$  collected by  $MA$

```

1 Initialize  $T$  as an empty set of data collected by  $MA$ 
2 while  $MA$  has not completed the allocated tasks do
3   Move to visit the next node  $n$  according to the given group  $gs$ 
4   if  $n$  has been visited by any other mobile agent then
5     Repeat Step 3, until all nodes in  $r$  have been visited
6   if all nodes in  $r$  have been visited then
7      $MA$  completes the allocated tasks
8   else
9     Dispatch  $MA$  to visit node  $n$ 
10    Collect data  $T'$  (not exceeding limitation  $d$  in total) from node  $n$ 
11    Add data in  $T'$  to  $T$ 
12    Set visited flag of node  $n$  to true
13  if  $T$  contains  $d$  data then
14     $MA$  completes the allocated tasks
15 Return  $T$ 

```

---

## 4 Experiments, evaluation and analysis

### 4.1 Simulation setup

We have implemented and tested the proposed GSMIP approach as well as existing approaches from the literature, namely SMIP [25], GIGM-MIP [21], and CL-MIP [19], using the Pymote simulator [28].

Pymote focuses on WSNs, which generally are networks of low power embedded devices. It is widely used by many researchers and developers to test distributed algorithms. The network model is adopted from [19] where 100 sensor nodes are uniformly deployed in an area and the sink node is placed at the centre of the area. Table 2 lists all of the mobile agent parameters used during simulation.

**Table 2** Simulation parameters of the proposed GSMIP approach

Network size	1000 × 500
Number of sensor nodes	100
Raw data size	2048 bits
MA code size	1024 bits
Data processing rate	50 Mbps
Raw data reduction ratio	0.8
Aggregation ratio	0.9
Radio transmission range	60 m
MA accessing delay	10 ms

**Table 3** Performance metrics for experimental work

Performance metrics Evaluation metrics	Definition
Energy consumption	Refers to the energy spent for transmitting, and receiving messages by mobile agent from all sensor nodes.
Task duration	Refers to the average time when the mobile agents are dispatched by the sink to the time when the last mobile agent returns back to the sink
Dispatched mobile agent	Refers to the number of dispatched mobile agent to collect data from all sensor nodes

## 4.2 Evaluation and analysis

To evaluate performance of different approaches, the following three performance metrics are considered: Task Duration, Energy Efficiency and Number of Dispatched Mobile Agents. Table 3 describes these three performance metrics.

*Energy Efficiency* As shown in Fig. 5, it is clear that our proposed GSMIP approach outperforms SMIP, GIGM-MIP, and CL-MIP approaches in terms of energy consumption. More energy is required for more agents to perform tasks in all of the four approaches. But we can observe that the proposed GSMIP exhibits better energy saving over other approaches. The proposed GSMIP approach achieves 31.2% and 13.3% energy decrease when compared to SMIP 36.4% and 15.2%, GIGM-MIP 47.4% and 17.1%, and CL-MIP 48.5% and 25.6% when the number of nodes decreases from 100 to 10. The improvement of the GSMIP approach on energy consumption is due to the fact that the number of hops for each MA is minimised within the groups.

*Task Duration* Fig. 6 compare the four approaches in terms of task duration. It is observed that our proposed GSMIP approach outperforms the three existing approaches, including SMIP, GIGM-MIP and CL-MIP. We can observe that the proposed GSMIP achieves the best task duration of all approaches. The proposed GSMIP achieves 50.8% and 20.4% task duration decrease when compared to SMIP 54.9% and 28.7%, GIGM-MIP 56.3% and 32.5%, and CL-MIP 65.9% and 40.2%, which has the

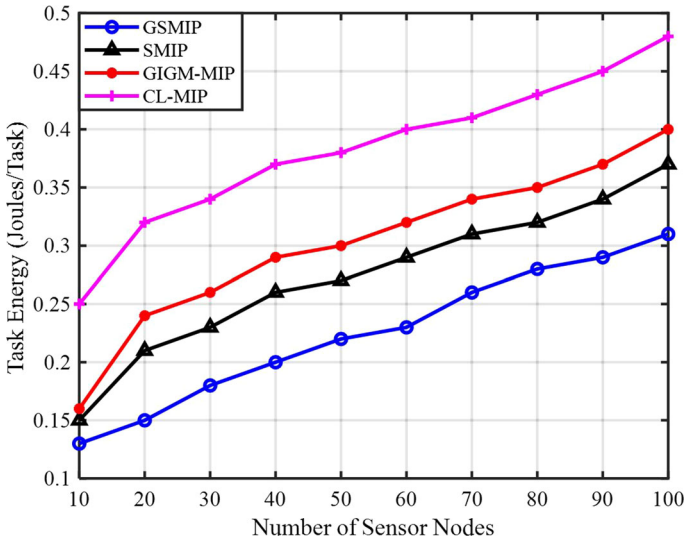


Fig. 5 The impact of number of sensor nodes on consumed energy

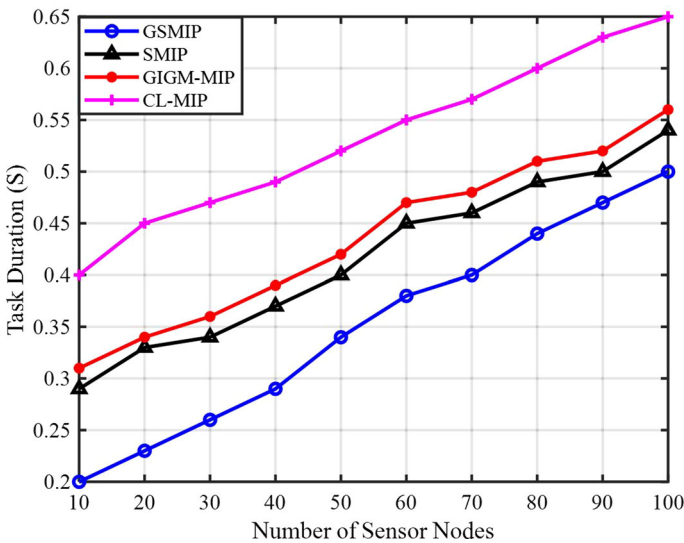


Fig. 6 The impact of number of sensor nodes on task duration

highest delay when the number of nodes decreases from 100 to 10. The improvement of the GSMIP for the task duration is due to the fact that the shortest itineraries are constructed for each MA in each group.

*The Effect of the Number of Dispatched Mobile Agents on Task Duration* Fig. 7 demonstrates the impact of the number of dispatched MAs on task duration for the proposed GSMIP. From Fig. 7, it is observed that when 20 mobile agents are dispatched

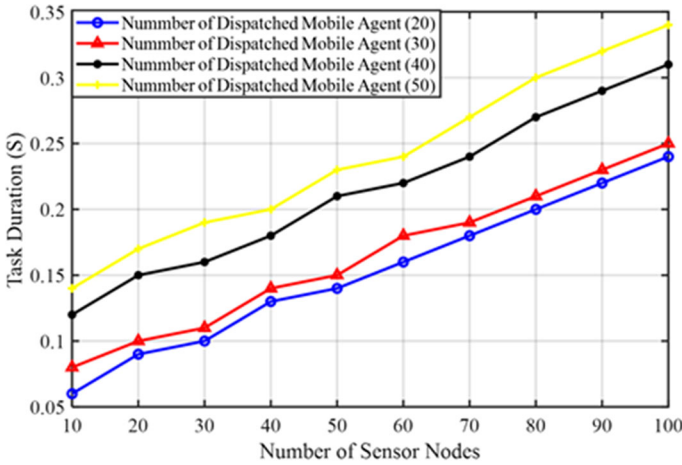


Fig. 7 The impact of number of dispatched mobile agent’s on task duration (MAs 20, 30, 40, 50)

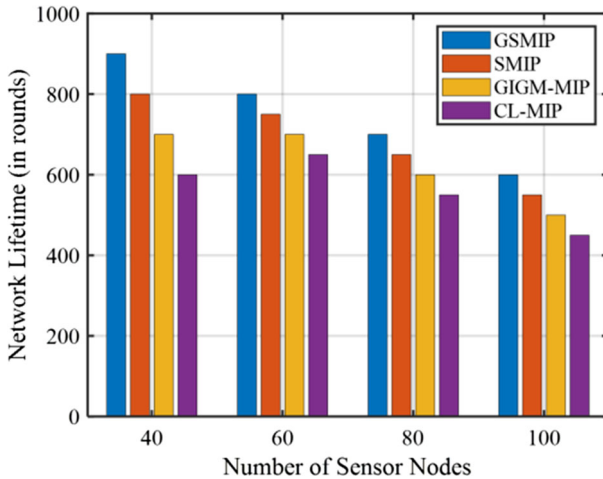


Fig. 8 The impact of number of sensor nodes on the network lifetime

and the number of nodes is 100, the task duration reaches 0.34/S; whereas when 30 mobile agents are dispatched and the number of sensor nodes is 100 the task duration reaches 0.27/S. Also, It is clear that when 40 mobile agents are dispatched and the number of sensor nodes is 100 the task duration reaches 0.24/S whereas when 50 mobile agents are dispatched and the number of sensor nodes is 100 the task duration reaches 0.22/S.

*Network Lifetime* Fig. 8 shows the impact of the number of sensor nodes on the network lifetime. It can be observed that as the number of nodes increases, the network lifetime for the proposed GSMIP is almost the same as compared to SMIP, GIGM-MIP, and CL-MIP that shows a noticeable decrease. This is due to the fact that the proposed GSMIP approach applies data aggregation and therefore carries a smaller



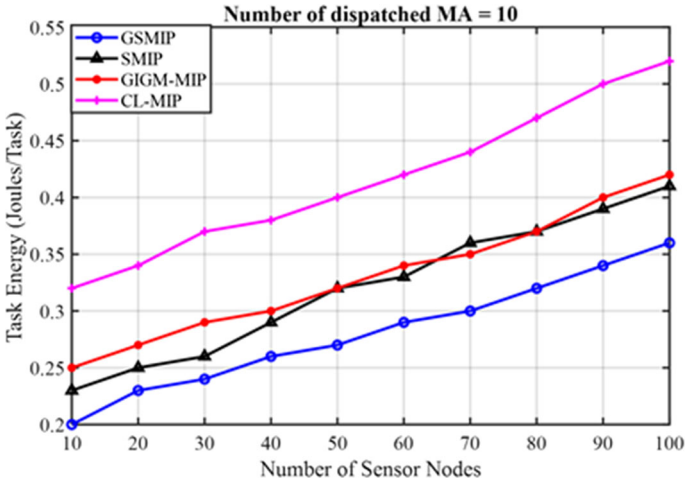


Fig. 9 The impact of number of dispatched mobile agent's on energy consumption (10 MAs)

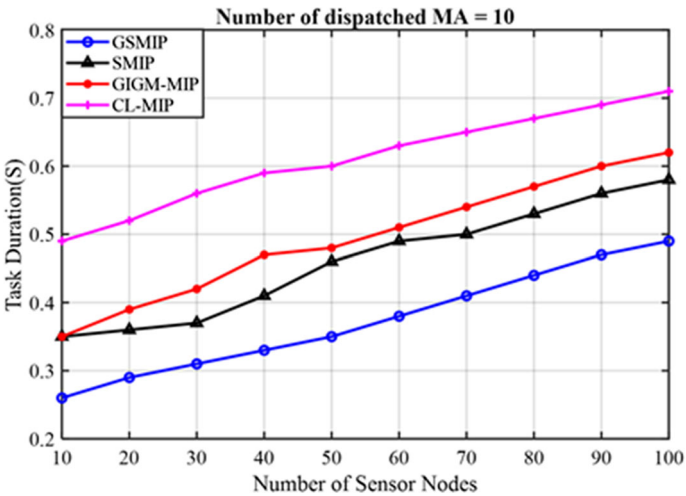


Fig. 10 The impact of number of dispatched mobile agent's on task duration (10 MAs)

size of the data packets, which leads to less energy consumption thereby increasing network lifetime.

*Energy Efficiency* As shown in Fig. 9, which shows the effect of the number of dispatched MAs on energy consumption. It is clear that our proposed GSMIP approach outperforms SMIP, GIGM-MIP, and CL-MIP approaches in terms of energy consumption while varying the number of dispatched MAs. This experiment considers the number of dispatched MAs as 10. We can observe that the proposed GSMIP exhibits better energy saving over other approaches. The proposed GSMIP approach achieves 36.3% and 20.1% energy decrease when compared to SMIP 41.2% and 24.1%, GIGM-

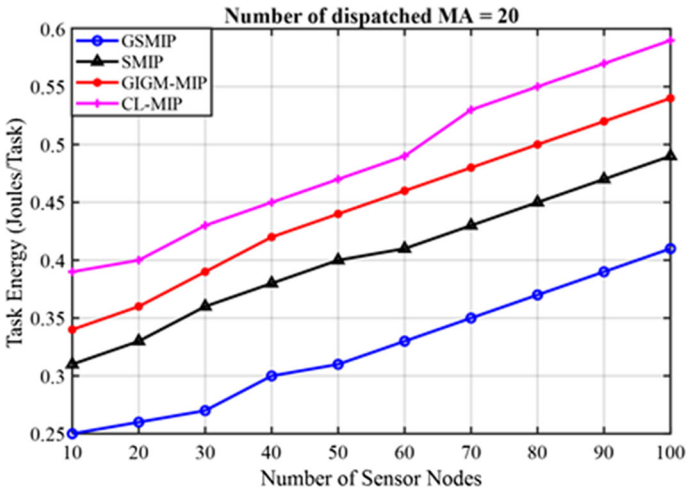


Fig. 11 The impact of number of dispatched mobile agent's on energy consumption (20 MAs)

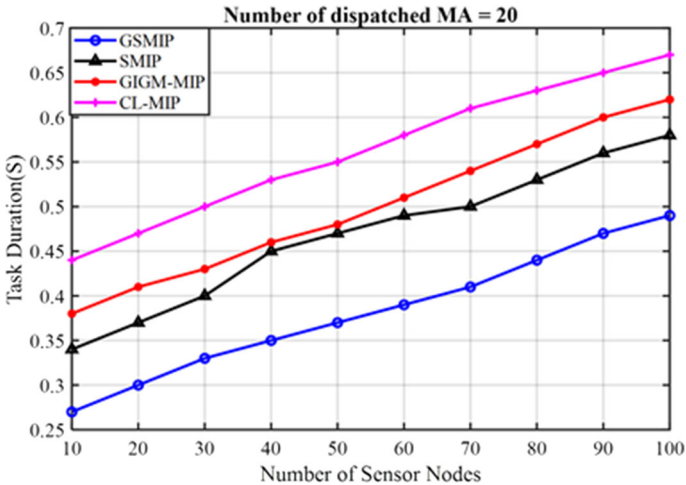


Fig. 12 The impact of number of dispatched mobile agent's on task duration (20 MAs)

MIP 43.4% and 25.1%, and CL-MIP 53.4% and 32.7% when the number of nodes decreases from 100 to 10. The improvement of the GSMIP approach on energy consumption is due to the fact that the number of hops for each MA is minimised within the groups.

*Task Duration* Fig. 10 compare the four approaches in terms of task duration while varying the number of dispatched MAs. The experiment considers the number of dispatched MAs as 10. It is observed that our proposed GSMIP approach outperforms the three existing approaches, including SMIP, GIGM-MIP and CL-MIP. We can observe that the proposed GSMIP achieves the best task duration of all approaches. The proposed GSMIP achieves 49.2% and 22.6% task duration decrease when compared

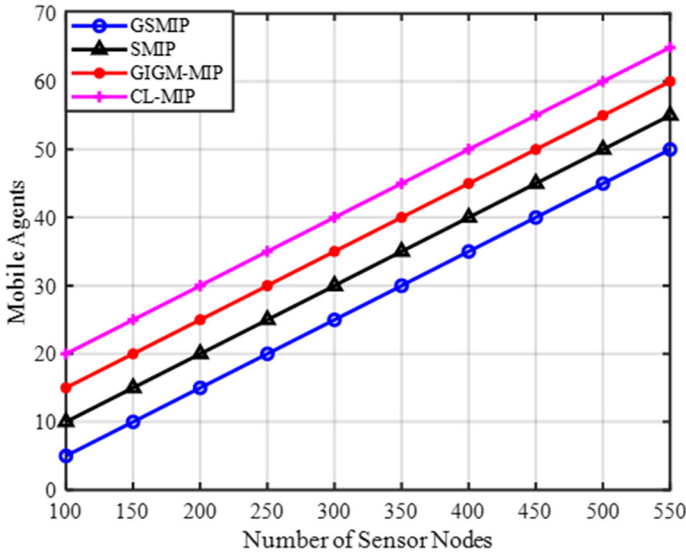


Fig. 13 The number of dispatched mobile agent of our proposed approach and alternative approaches

to SMIP 59.8% and 32.9%, GIGM-MIP 61.2% and 32.5%, and CL-MIP 71.5% and 49.3%, which has the highest delay when the number of nodes decreases from 100 to 10. The improvement of the GSMIP approach for the task duration is due to the fact that the shortest itineraries are constructed for each MA in each group.

*Energy Efficiency* As shown in Fig. 11, which shows the effect of the number of dispatched MAs on energy consumption. It is clear that our proposed GSMIP approach outperforms SMIP, GIGM-MIP, and CL-MIP approaches in terms of energy consumption while varying the number of dispatched MAs. The experiment considers the number of dispatched MAs as 20. We can observe that the proposed GSMIP exhibits better energy saving over other approaches. The proposed GSMIP approach achieves 41.4% and 25.2% energy decrease when compared to SMIP 49.6% and 31.3%, GIGM-MIP 57.5% and 34.2%, and CL-MIP 59.6% and 40.8% when the number of nodes decreases from 100 to 10. The improvement of the GSMIP approach on energy consumption is due to the fact that the number of hops for each MA is minimised within the groups.

*Task Duration* Fig. 12 compare the four approaches in terms of task duration while varying the number of dispatched MAs. The experiment considers the number of dispatched MAs as 20. It is observed that our proposed GSMIP approach outperforms the three existing approaches, including SMIP, GIGM-MIP and CL-MIP. We can observe that the proposed GSMIP achieves the best task duration of all approaches. The proposed GSMIP achieves 50.3% and 26.5% task duration decrease when compared to SMIP 56.6% and 34.8%, GIGM-MIP 63.3% and 37.4%, and CL-MIP 67.3% and 45.4%, which has the highest delay when the number of nodes decreases from 100 to

10. The improvement of the GSMIP approach for the task duration is due to the fact that the shortest itineraries are constructed for each MA in each group.

*Number of Dispatched Mobile Agent* Fig. 13 shows the different number of dispatched mobile agents for each approach. From Fig. 13, we can see that CL-MIP dispatches the highest number of mobile agents, which is more than 60 MAs whereas GIGM-MIP dispatches 60 MAs followed by SMIP that dispatches more than 50 MAs. In contrast, our proposed GSMIP dispatches the minimum number of MAs, which is 50 MAs.

## 5 Conclusion and future work

This paper has addressed the issue of efficient multi-mobile agents itinerary planning. In order to achieve scalability and to reduce energy consumption, an energy-efficient itinerary planning approach, called GSMIP, has been proposed. A grouping strategy is developed where nodes in the network are assigned into smaller sets, making energy depletion less of a problem. Experimental evaluation shows that the proposed itinerary planning approach is scalable, energy-efficient, and reduces delays while increasing network lifetime.

There are a number of interesting directions for future work. We plan to investigate the possibility of driving a dynamic or a hybrid itinerary planning for MAs, which allows each MA to decide the visiting sequence on-the-fly. This is particularly useful for providing fault-tolerance and can be achieved by adopting an efficient clustering method in which nodes can be grouped according to specific criteria, and eventually MAs will be directed to a particular group as described in [2].

## References

1. Alsboui T, Abuarqoub A, Hammoudeh M, Bandar Z, Nisbet A (2012) Information extraction from wireless sensor networks: system and approaches. *Sensors Transducers* 14(1–17):03
2. Alsboui T, Alrifaae M, Etaywi R, Jawad MA (2016) Mobile agent itinerary planning approaches in wireless sensor networks- state of the art and current challenges. In: Maglaras LA, Janicke H, Jones KI (eds) *Industrial networks and intelligent systems—second international conference INISCOM 2016*, Leicester, UK, October 31–November 1, 2016, Revised Selected Papers, Lecture notes of the Institute for Computer Sciences, Social informatics and telecommunications engineering, vol 188, pp 143–153. [https://doi.org/10.1007/978-3-319-52569-3\\_13](https://doi.org/10.1007/978-3-319-52569-3_13)
3. Pottie GJ, Kaiser WJ (2000) Wireless integrated network sensors. *Commun ACM* 43(5):51–58
4. Wu C, Rao N, Barhen J, Iyengar S, Vaishnavi V, Qi H, Chakrabarty K (2004) On computing mobile agent routes for data fusion in distributed sensor networks. *Knowl Data Eng IEEE Trans* 16(740–753):07
5. Chen G, Wu S, Zhou J, Tung AKH (2014) Automatic itinerary planning for traveling services. *IEEE Trans Knowl Data Eng* 26(3):514–527
6. Chen Min, Kwon Taekyoung, Yong Yuan, Victor Leung (2006) Mobile agent based wireless sensor networks. *J Comput* 1:04
7. El Fissaoui M, Beni-Hssane A, Ouhmad S, El Makkaoui K (2020) A survey on mobile agent itinerary planning for information fusion in wireless sensor networks. *Arch Comput Methods Eng* 03
8. Qi H, Wang F (2004) Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks. 02

9. Bo L, Jiuxin C, Jie Y, Wei Y, Benyuan L, Xinwen F (2016) Disjoint multi mobile agent itinerary planning for big data analytics. *EURASIP J Wirel Commun Netw* 2016:12
10. Junfeng W, Yin Z, Zhuanli C, Xuan Z (2015) Emip: energy-efficient itinerary planning for multiple mobile agents in wireless sensor network. *Telecommun Syst* 62:03
11. Yu-Cheng C, Madoka N (2017) A clonal selection algorithm for energy-efficient mobile agent itinerary planning in wireless sensor networks. *Mobile Netw Appl* 23:01
12. Chen M, Kwon T, Yong Y, Choi Y (2007) Mobile agent-based directed diffusion in wireless sensor networks. *EURASIP J Adv Signal Process* 2007:01
13. Jiang F, Shi H, Xu Z, Dong X (2009) Improved directed diffusion-based mobile agent mechanism for wireless sensor networks. In: 2009 4th International conference on communications and networking in China, CHINACOM 2009. 08
14. Damianos G, Ioannis V, Charalampos K, Grammati P (2016) Energy-efficient multiple itinerary planning for mobile agents-based data aggregation in WSNS. *Telecommun Syst* 63:02
15. Dong M, Ota K, Yang Laurence T, Chang S, Zhu H, Zhou Z (2014) Mobile agent-based energy-aware and user-centric data collection in wireless sensor networks. *Comput Netw* 74:58–70 Special issue on mobile computing for content/service-oriented networking architecture
16. Min C, Laurence Y, Ted K, Liang Z, Minhho J (2011) Itinerary planning for energy-efficient agent communications in wireless sensor networks. *Veh Technol IEEE Trans* 60(3290–3299):10
17. Tariq A, Yongrui Q, Richard H, Hussain A-A (2020) Enabling distributed intelligence for the internet of things with IOTA and mobile agents. *Computing* 102(6):1345–1363
18. Chen M, González-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
19. Chen M, Gonzalez S, Zhang Y, Leung VCM (2009) Multi-agent itinerary planning for wireless sensor networks. In: Bartolini N, Nikolettseas S, Sinha P, Cardellini V, Mahanti A (eds) Quality of service in heterogeneous networks. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 584–597
20. Mohamed El Fissaoui, Beni Hssane A, Saadi M (2018) Multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks. *EURASIP J Wirel Commun Netw* 92:12
21. Imene A, Okba K, Laid K, Sylvie S (2015) A new itinerary planning approach among multiple mobile agents in wireless sensor networks (WSN) to reduce energy consumption. *Int J Commun Netw Inf Secur (IJCNIS)* 7(116–122):08
22. Govind G, Manoj M, Kumkum G (2017) Towards scalable and load-balanced mobile agents-based data aggregation for wireless sensor networks. *Comput Electr Eng* 64:10
23. Egwuche OS, Adewale OS, Oluwadare SA, Daramola OA (2020) Enhancing network life-time of wireless sensor networks through itinerary definition and mobile agents for routing among sensor nodes. In: 2020 International conference in mathematics, computer engineering and computer science (ICMCECS), pages 1–7
24. Govind G, Manoj M, Kumkum G (2014) Energy and trust aware mobile agent migration protocol for data aggregation in wireless sensor networks. *J Netw Comput Appl* 41:05
25. Huthiafa Q, Zuriati Z, Hanapi ZM, Shamala S (2017) A spawn mobile agent itinerary planning approach for energy-efficient data gathering in wireless sensor networks. *Sensors* 17(1280):06
26. Tseng Y-C, Kuo S-P, Lee H-W, Huang C-F (2003) Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. In: Feng Z and Leonidas G (eds) Information processing in sensor networks, pp 625–641. Springer, Berlin, Heidelberg
27. Chen M, Kwon T, Yuan Y, Choi Y (2007) Mobile agent-based directed diffusion in wireless sensor networks. *EURASIP J Adv Signal Process* 1:036871
28. Damir A, Kristijan L (2013) Pymote: high level python library for event-based simulation and evaluation of distributed algorithms. *Int J Distrib Sensor Netw* 9(3):797354