Article

A Dynamic Multi-Mobile Agent Itinerary Planning Approach in Wireless Sensor Networks via Intuitionistic Fuzzy Set

Tariq Alsboui 1,*, Richard Hill 1, Hussain Al-Aqrabi 1, Hafiz Muhammad Athar Farid 2, Muhammad Riaz 2, Shamaila Iram 1, Hafiz Muhammad Shakeel 1 and Muhammad Hussain 1

1 Department of Computer Science, School of Computing and Engineering, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK
2 Department of Mathematics, University of the Punjab, Lahore 54590, Pakistan
* Correspondence: t.alsboui@hud.ac.uk

Abstract: In recent research developments, the application of mobile agents (MAs) has attracted extensive research in wireless sensor networks (WSNs) due to the unique benefits it offers, such as energy conservation, network bandwidth saving, and flexibility of open usage for various WSN applications. The majority of the proposed research ideas on dynamic itinerary planning agent-based algorithms are efficient when dealing with node failure as a result of energy depletion. However, they generate inefficient groups for MAs itineraries, which introduces a delay in broadcasting data return back to the sink node, and they do not consider the expanding size of the MAs during moving towards a sequence of related nodes. In order to rectify these research issues, we propose a new Graph-based Dynamic Multi-Mobile Agent Itinerary Planning approach (GDMIP). GDMIP works with “Directed Acyclic Graph” (DAG) techniques and distributes sensor nodes into various and efficient group-based shortest-identified routes, which cover all nodes in the network using intuitionistic fuzzy sets. MAs are restricted from moving in the predefined path and routes and are responsible for collecting data from the assigned groups. The experimental results of our proposed work show the effectiveness and expediency compared to the published approaches. Therefore, our proposed algorithm is more energy efficient and effective for task delay (time).

Keywords: wireless sensor networks; mobile agent; itinerary planning; dynamic itinerary; intuitionistic fuzzy set; human–robot interaction

1. Introduction

The ability to perform fundamental functions by sharing resources within a network and node domains characterises a prevalent network that is interconnected, also expanding to wireless sensor networks (WSN).

WSNs’ main objective is to enable users to acquire relevant information from data collected by spatially distributed sensors. To have complete view of the controlled physical environment, many sensors are typically deployed. Large amounts of data will be created by these sensor network systems, as stated in [1]. Effective collective data collection from (wireless sensor networks) WSNs has been made possible by the widespread deployment of MA techniques. In these methods, MAs are sent out to traverse the sensor network following predetermined routes that are established via itinerary planning in order to collect data from the related nodes along the route. The necessity to find and handle MAs in the energy-efficient systems of WSN, which is essentially defined by methods for designing MA itinerary planning.

Itinerary planning is challenging due to practical restrictions for the deployment of complex nodes and networks, for instance, processing capabilities and the limitation of battery power. The movement cost of a MA, route planning, and the techniques deployed for the designing of such a plan are essential aspects when transmitting MAs to the network.
The MA’s key aim is to collect, use, and process collected data within a network. They can come together and decide locally on their own without user input. Because radio communication is one of the most efficient hungry activities, mobile agents are typically utilised [2]. We, therefore, send agents with the aim of procuring data despite transmitting it towards a sink node to prevent long-distance radio communication. Planning the route of mobile agents in such cases is essential to reducing sensor node energy usage. However, it has proved difficult to tackle the “NP-hard” problem of determining an optimal route for MAs to visit sensor nodes sequence [3,4].

A significant amount of research has been performed over the past few decades to produce human-like gaze control in an effort to improve natural human–robot interaction (HRI). This endeavour is part of a larger effort to increase HRI [5]. The fundamental idea behind HRI is that the robot should behave in a manner similar to that of an animal friend to a human. According to this paradigm, the robot should neither be shaped to copy the human person nor should it develop communication similar to that which occurs between humans; rather, it should follow the models set by biological systems already in place and form an interaction between different species [6]. This study does not cover systems that use fuzzy logic for sensing, modelling, or planning activities; rather, it only considers systems that use fuzzy logic for inferencing or reasoning tasks. This is because fuzzy logic has already been widely employed in practically all research fields in robotics [7].

The development of an acceptable itinerary for (mobile agents) MAs to gather the related data from all sensors is thus a major task [8]. Planning an itinerary is the process of deciding the path an MA uses when navigating and visiting the network to collect data from the nodes. Each route involves a list of related connected nodes that will stay linked throughout the MA movement journey. The existing techniques can be largely divided into three classifications for establishing MA itineraries: the first is a static itinerary, the second one is a dynamic itinerary, and the third is a hybrid itinerary [8]. In a static itinerary, the dispatcher, or sink node, specifies the order of the nodes to that the mobile agent will pay a visit. In the dynamic itinerary, MAs change the path of the visit on the fly to avoid node failure in the network, whereas, in the hybrid itinerary, the list of sensor nodes to be visited is generated by the sink node, and is changeable on the fly.

Contribution: Our research focuses on the development of a Graph-based Dynamic Multi-Mobile Agent Itinerary Planning approach (GDMIP) that utilises Directed Acyclic Graph (DAG) and intuitionistic fuzzy set in the planning as well as dispatching of mobile agents.

The contributions of this paper are presented as:

- Dynamic multi-mobile agent (MMA) itinerary planning approach for managing resources in a more energy-efficient and scalable manner.
- A novel approach, dubbed GDMIP. GDMIP computes mobile agents itinerary planning based on the “Directed Acyclic Graph” (DAG) technique and selects the shortest route encompassing all the assigned nodes in the proposed network using intuitionistic fuzzy sets. It segments DAG into small groups and allocates mobile agents (MAs) to distinct groups to be more energy efficient.
- In the end, we conducted several experiments to validate the effectiveness and usefulness of the proposed dynamic approach.

The structure of the remaining research paper is presented as follows: Section 2 presents the previous related work providing a segue into our work and contribution. Section 3 describes our methodology. In Section 4, an in depth discussion of the benefits of our framework, along with a comparison with similar and competing methodologies. Lastly, in Section 5, we present our conclusions along with future research directions.

2. Related Work

The majority of published work found in the literature assumes that an MA has a lack of prior knowledge of their path when it comes to dynamic itinerary planning in the wireless sensor network for mobile agents. They also employ MA to gather information.
A dynamic “Energy and Trust Aware Mobile Agent Migration” (ETMAM), for instance, is suggested in [9]. The primary goal of ETMAM is to locate and avoid hostile or defective nodes when migrating mobile agents. In order to design routes quickly for the travelling agent to carry out the data aggregation activities, ETMAM combines energy and trust as selection criteria. The suggested method is secure, responds to network node failures effectively, and uses little energy. Additionally, ETMAM optimises the agent migration path and lessens the agent payload through the usage of cloning mechanisms. When visiting sensor nodes, ETMAM, however, does not account for the MA’s expanding size. Additionally, it causes a delay in data reporting because the cloned MAs still need to visit a lot of nodes. Furthermore, the entire strategy is regarded as difficult and necessitates considerable processing due to the requirement of recognising malicious nodes.

A dynamic technique of constructing a mobile agent path with a minimal payment based on referral was proposed in [10]. Referral is a cooperative approach for discovering the solution with assistance, presuming that the person asking the question is unaware of the solution but is aware of those who can assist him. Therefore, based on his acquaintance’s knowledge, the present workplace provider can suggest the next workplace (host) for a mobile agent. The suggested method can adapt to an open environment because it is energy-efficient and does not require a pre-fixed system. However, the method lacks security and scalability. Additionally, because it only employs one MA, a delay is introduced.

In [11], a dynamic itinerary planning method based on the Mobile Agent Electronic Triage Tag (MAETT) system is suggested. In order to provide a straightforward system for flexibly defining and tracking an agent’s journey, the aim is to decouple the itinerary data structure from the agent itself. The suggested strategy is scalable and energy efficient. Additionally, it responds effectively when a node fails. However, the method is insecure and requires that the injecting node, which is the one connecting to the portable device, be kept inside the WSN’s connection range. Because of this, medical personnel are required to employ that node as a component of the WSN, replacing the injecting node each time a new WSN is generated. Refs. [10,12] suggests similar strategies. By providing a dynamic planning technique for a mobile agent path based on the Markov decision process (MDP) and social acquaintance recommendation variable decision space in [10], the authors expand on their earlier work. The path planning model is explained with MDP, social acquaintance recommendations, and the assumption that each social member can cooperatively recommend a workplace and service to a mobile agent. The method uses less energy and reacts to node failure quickly. However, it takes a lot of time and effort to actually use their method. Additionally, it causes a delay, lacks security, and is not scalable.

A different strategy is one that is suggested in [13], where the authors focus on the computing side of itinerary planning to let an agent complete its task while adhering to a deadline. A subset of resources that the agent must retrieve from network servers while adhering to a partial order defines the mission. The term “resource” refers to an ambiguous idea that could, among other things, refer to a device, a database, or processing power. The network servers offer resources of various types and quality, so choosing which servers to visit first and in what sequence to do so will help complete the job on time and with the highest quality possible. By establishing a graph-theoretic model for itinerary computation within time constraints, the work presented in [12] advances the work on computation itineraries. This model enables the creation of provably optimum, dynamic programming (DP), and approximation algorithms. These methods respond to node failure effectively and use little energy. They do not, however, offer security support, and the lengthy routes for a particular set of MAs cause a delay due to the inefficient grouping of sensor nodes.

Dynamic itinerary planning for MAs (DIPMA) is presented in [14]. With an application-oriented methodology, it gathers data from sensor networks. To be more precise, a real-world application of the DIPMA method is the data collection for frost prediction. The DIPMA operates in accordance with two phases, including feature vector design policy and next-hop determination. Instead of delivering an MA across numerous nodes, the design policy of feature vectors swaps vectors between nodes. The concept of choosing the next
The hop of MA migration using feature vectors is the subject of the next hop choice. The method takes less time to execute and is scalable and energy efficient. However, it only applies to a certain application because searching for an $n$-hop depth results in creating data at a depth that is not optimal. Further, it introduces a delay in reporting data back due to the fact the MAs are required to visit a large number of nodes in each group, and lacks support for security.

The authors of [15] suggest “spawn multi-mobile agent itinerary planning” (SMIP) to greatly minimize the considerable increases in energy expenditure and time required for data collecting. The spawning mobile agent, which enables the essential MA to set off another MA into a single segment, serves as the foundation for this. Based on Bayesian assessments, the proposal determines the number of clusters using x-means methods. When partitioning is finished, the sink node specifies the number of MAs and their travel routes for each partition. Although the method is scalable, it does not account for node failure in the network, which the authors propose as a future area of study.

A three-phased multiple mobile agent itinerary planning method called GIGM-MIP is introduced in [16]. Using the $k$-means algorithm and a set of produced partitions based on geographic information, the network is divided in the first phase. Each partition may contain a number of mobile agents. The second phase counts the number of mobile agents and defines groups of nodes for each agent. The third phase is focused on determining the route that each mobile agent takes while travelling through the cluster of source nodes. Each partition may get a number of mobile agents. The GIGM-MIP technique is scalable and energy-efficient. However, it does not take into consideration node failure in the network, creates a delay, and does not provide security.

In [17], a novel “central location-based MIP” (CL-MIP) framework was developed. The “visiting central location” (VCL) selection algorithm, SIP method, source-grouping algorithm, and its iterative algorithm are the four components that make up the framework. The high source node density is computed using the VCL selection algorithm. Nodes are grouped, and mobile agents are assigned to specific groups via the source-grouping algorithm. To plan the route of mobile agents, the SIP algorithm is used. Finally, the primary goal of the iterative process is to guarantee that all source nodes are assigned to the designated MAs. The CL-MIP method takes into account a cluster-based approach in which the source nodes are deployed across various clusters and grouped geographically. This shows that if the nodes are present, the CL-MIP cannot be employed if the nodes are sparsely deployed.

The authors of [18] suggested a system called MAMS that uses a mobile server and agent to gather data from sensor nodes placed throughout a sensing field. After gathering data, mobile agents independently move between nodes before returning to the mobile server. In order to route mobile agents, the migration process uses a geographic routing strategy. After gathering data, mobile agents locate the mobile server’s current location and head there with the compiled data. The system’s key component is an efficient and perceptive gathering method. However, it introduces a delay due to inefficient grouping, as the mobile agents will have to visit a large number of nodes.

In [19], a new dynamic itinerary planning method is proposed. The three phases that make up SLMADA suggested scheme’s operation are (1) network setup, (2) zone coordinator selection, and (3) MA migration phase, each of which is in charge of a different duty. The strategy is scalable and energy-efficient. Additionally, load-balancing for mobile agent-based data aggregation is accomplished. The performance of the suggested agent migration scheme under cognitive radio sensor networks may be subpar, as the authors note as future work. However, designing a sleep schedule for the nodes that are visited by the mobile agents is crucial, and this could affect how well the proposed agent migration scheme performs. Lengthy MAs routes may cause a delay in reporting data and a lack of support for security.

Jana et al. proposed a “multi-attribute decision making” (MADM) approach using power Dombi aggregation operators (AOs) [20], a dynamical hybrid approach to design the
decision-making process [21], and the MADM technique based on complex AOs [22]. Farid and Riaz proposed some generalized Einstein-interactive geometric AOs [23], Prioritized Interactive AOs [24] for q-rung orthopair fuzzy (q-ROF) information. Riaz et al. proposed prioritized AOs based on priority degrees for q-ROF data [25]. There is some brilliant work related to MADM seen in [26–28]. Ashraf et al. introduced some AOs for cubic picture fuzzy sets [29], symmetric sum-based AOs for spherical fuzzy data [30]. Ali et al. gave the notion of complex intuitionistic fuzzy soft AOs [31]. Novel score functions related to q-ROF data are given by Feng et al. [32]. Alcantud et al. [33] proposed a MADM method based on N-soft set. Deveci et al. [34] proposed a Dombi-Bonferroni Operator-based decision-making model.

The authors of [35] introduce FuMAM, a fuzzy-based MA migration strategy, in their approach. By taking into account three factors, remaining energy, distance, and the number of neighbours, FuMAM determines the best itineraries for MA. The proposed use of the FuMAM approach results in an increase in the percentage of successful MA round trips. In addition, the FuMAM algorithm extends the amount of time a network can function by selecting the node in the network that has the highest amount of residual energy as the next hop for MA migration. However, owing to the lengthy MA itineraries, there is a delay introduced.

The FuMAM-based approach is similar to the GDMIP approach proposed in this paper. Both approaches utilise fuzzy logic. However, our approach focuses on intuitionistic fuzzy sets, and we consider different criteria for the MA itinerary, such as grouping, energy efficiency, least number of nodes, and avoiding excessive delay. These criteria are considered efficient in determining the itinerary planning of each MA.

### 3. Proposed GDMIP

#### 3.1. GDMIP Architecture

The thematic demonstration of our proposed GDMIP approach is shown in Figure 1 (GDMIP). All important elements are described, including itineraries, MAs, nodes, faulty nodes, and gathered data, and each component works its own certain function. It displays the arrangement of the related nodes with the specified route. The discovered shared nodes form the basis of the grouping. All network nodes are covered by the routes that are created. The sink node is in charge of allocating MAs to a specific group to gather data from. If a node is faulty in the group as a result of energy depletion during MA migration, the MA will pass that node in the group. The MA moves and collects data only from the groups that have been assigned to it. As an illustration, the path to the designated group is shown by the “itinerary” (orange line circles).

We employ a number of MAs to prevent reporting delays and to promote local interactions. We use a methodology similar to that described in [36,37] to determine the volume of sensory data that MAs should gather. In order to eliminate duplication and inconsistency, we use equivalent data aggregation techniques. A combination of sensory data effects and the ratio of an aggregation (“$\rho$, $0 \leq \rho \leq 1$”) is used.
Figure 1. The Proposed Dynamic Approach for Multi-Mobile Agents Itinerary Planning.

$L_{ma}^i$ is the quantity of the related sensor data for aggregation via the fusion factor $\rho$, with $A_i$ having the actual quantity of related sensor data after the mobile agent receives data input from the source node $i$.

\[
\begin{align*}
L_{ma}^i &= A_i \\
L_{ma}^2 &= A_i + (1 - \rho) \times A_2 \\
L_{ma}^i &= L_{ma}^i + (1 - \rho) \times A_2 \\
&= A_1 + \sum_{g=2}^{1} (1 - \rho) \times A_g
\end{align*}
\] (1)

Equation (3) indicates that there is no data aggregation for the first node; also the value of $\rho$ depends on the type of application being used.

Figure 2 defines the proposed GDMIP data packet format. The initial data collection from the source node and the MA are represented by FirstNode. We have established groups and routes. A group merely decides which nodes the MAs should collect data from. However, all computed routes are constructed with MAs migrating to every node on the route.

The agents’ payload consists of the messages Itinerary Planning and List of data. The node at the root that posts and generates the current mobile agent is recognized by the Dispatcher ID. ToVisitFlag is set up to check whether an agent visited the node or not.
In order to gather data from the nodes along this specific path, routes are computed for the mobile agents to travel towards and visit each node and the group of nodes.

**MAAs Payload**

![Itinerary Planning](image1)

![List of Data](image2)

**Immutable Attribute**

| Dispatcher ID | FirstNode | ToVisitFlag |
|---------------|-----------|-------------|

*Figure 2. Proposed itinerary message format.*

Additionally, we are of the view that sensors are expected to generate similar data when placed close together in order to simulate a real-world scenario. In light of this, agents may remove unnecessary data through fusion.

### 3.2. GDMIP Algorithms

An explanation of the algorithms utilised in the suggested GDMIP approach is provided in this section.

For the purpose of constructing simulation situations, Algorithm 1 is supplied with the pseudocode to produce an arbitrary DAG $G$. 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. The multi-mobile agent’s dispatched pseudocode is shown in Algorithm 4, which was used to commence the collection of sensory data.

**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. 
4. 
5. **while** $node_b == node_a$ **do**
6. 
7. Add edge($node_a$, $node_b$) to $G$
8. **if** $G$ is still DAG **then**
9. 
10. **else**
11. 
12. Get DFS post-route ($G$)
13. Return DFS post-order
14. Return $G$

First, we create a random-sized network with a random distribution of nodes and edges using a random DAG, as described in Algorithm 1. The required amount of $nodeNum$ nodes is added by the iterated method. The $G$ graph is then positioned to examine if it is
completely linked with DAG—that is, if all the related source nodes are moved back after visiting the sink node. A deep first search is used to check and confirm if the relevant nodes have examined.

Communication links between nodes in a typical WSN deployment might be either symmetric or asymmetric. Because it will make our method more flexible, we have presumed asymmetric communication and network links on this occasion. If a communication channel is symmetric, our method can still be used by employing only that one link.

3.3. Intuitionistic Fuzzy Weighted Averaging (IFWA)

In multi-mobile agent itinerary planning algorithms, there are many factors involved. These factors are uncertain due to their nature. Fuzzy logic is similar to the natural decision-making process. It interacts with unclear and ambiguous knowledge. This is a simplified view of authentic situations, relying on stages of truth instead of the typical 1/0 or true/false as Boolean reasoning.

The phrase “fuzzy” refers to things that are unclear. We commonly meet circumstances in the real world in which we are unable to determine if such a condition is factually true, such as in this proposed approach, we are uncertain of the identified shortest route with the least number of nodes, in which fuzzy logic provides a highly useful variety of thinking. We may use this strategy to investigate the mistakes and inconsistencies in any circumstance. The Boolean paradigm represents 1.0 as the ultimate truth value and 0.0 as the ultimate false value. However, in the case of fuzzy systems, there is an intermediary value that is both partially true and most likely partially false. Lofti A. Zadeh invented fuzzy logic in his academic research “fuzzy sets (FSs)” in 1965 [38].

A FS $F$ in the universe $X$ can be characterised as a set of ordered pairs and formally depicted as

$$ F = \{(x, \mu_F(x)) | x \in X\} $$

here, $\mu_F(x)$ represents the “membership function (MF)” whose range is $[0, 1]$.

Fuzzy logics have various uses, such as altitude management of spaceships and rockets in the aviation industry. It has been utilised in automobile systems to manage speed and traffic, for decision-making assistance systems and security appraisal, for chemical process industries to manage the pH, drying, and chemical distillation processes, for natural language translation, and a variety of intense machine learning and artificial intelligence. Fuzzy logic is often employed in control design systems such as expert systems and in conjunction with Neural Networks because it simulates how humans make judgments, only much faster.

It is not always the case that the grade of non-membership of an object in an FS is equivalent to 1 minus the membership degree; this statement is not always accurate. As a result, intuitionistic fuzzy sets (IFSs) were proposed as a generalisation of FSs. The idea of defining IFSs as generalised FSs is intriguing and beneficial in a variety of application areas. Because it incorporates the “degree of belongingness”, “degree of non-belongingness”, and the hesitation margin, the information and semantic description of IFS become more significant, resourceful, and useful. A IFS $I$ in the universe $X$ can be characterised as a set and formally depicted as,

$$ I = \{(x, \mu_I(x), \nu_I(x)) | x \in X\} $$

here $\mu_I(x)$ represents the MF and $\nu_I(x)$ represents the non-MF; their ranges are $[0, 1]$, with the constraint that the $\mu_I(x) + \nu_I(x) \leq 1$, $\forall x \in X$ [39].

Data aggregation is a key step in the selection process across many domains, including corporate, legislative, social, academic, technological, intellectual, and intelligent machines. Throughout the course of human history, comprehension of options has traditionally been regarded as a discrete or spoken amount. In the most recent decades, there has been a significant focus placed on the consolidation of various types of information. The efficiency of the AO, as well as their constraints, have been ingrained in the decision-making process. It should come as no surprise that AO incorporates a variety of standard procedures for combining a limited number of fuzzy numbers into a single fuzzy number.
Many elements are included in itinerary planning algorithms, including (C₁) grouping, (C₂) energy efficiency, (C₃) least number of nodes, and (C₄) avoiding excessive delays. Because these characteristics are ambiguous and unpredictable, we employ intuitionistic fuzzy numbers (IFNs) to better explain the information. In Figure 3, as can be seen, there are three different routes (route 1, route 2, and route 3), and if we have a decision-maker (DM) to determine the best route based on the parameters (criteria) listed above, the DM assigns an IFN to each route based on its assessments of each criterion. We get a decision matrix after evaluating all paths and then use the “intuitionistic fuzzy weighted averaging (IFWA) operator” to aggregate the data [40]. In IFWA operators, each criterion has a weight vector (WV) that specifies its relevance. An IFWA operator defines if we have a collection of IFNs given as  \( \alpha_k = (\mu_k, \nu_k) \), where \( k = 1, 2, 3, \ldots, n \). The IFWA operator is a mapping from \( X^n \rightarrow X \) and is given as

\[
IFWA(\alpha_1, \alpha_2, \ldots, \alpha_n) = \left( 1 - \prod_{i=1}^{n} (1 - \mu_{\alpha_i})^{\omega_i}, \prod_{i=1}^{n} \nu_{\alpha_i} \right)
\]

where \( \omega = (\omega_1, \omega_2, \ldots, \omega_n) \) is the WV of \( \alpha_i \) with the constraint that the sum of WV is a unit.

A score function is used to define how one IFN is larger than another IFN. An IFN \( \alpha \) having a larger score value than another IFN \( \beta \), then \( \alpha \) is larger than \( \beta \). The score function is defined as

\[
S(\alpha) = \mu_\alpha - \nu_\beta.
\]

A DM (or group of DMs) evaluates the three routes (route 1, route 2, and route 3) on the basis of the four criteria, (C₁) grouping, (C₂) energy efficiency, (C₃) least number of nodes, and (C₄) avoiding excessive delays, given in Table 1. The WV of these criteria is \( (0.20, 0.25, 0.25, 0.30) \). Using the IFWA operator, we aggregate all IFNs for each route to obtain a single IFN for each route. Following that, we apply the scoring function to each aggregated IFN. If an aggregated IFN has a higher score value, the route associated with this IFN is a shorter route. As per the data given in Table 1, the best route is “route 2”. Algorithm 2 is given for the working of an IFWA operator for selecting the optimum route solution.

**Table 1.** Decision matrix acquired from DM.

|       | (C₁)       | (C₂)       | (C₃)       | (C₄)       |
|-------|------------|------------|------------|------------|
| route 1 | (0.70, 0.20) | (0.85, 0.10) | (0.75, 0.20) | (0.55, 0.15) |
| route 2 | (0.45, 0.30) | (0.50, 0.20) | (0.65, 0.15) | (0.30, 0.55) |
| route 3 | (0.35, 0.15) | (0.55, 0.25) | (0.25, 0.45) | (0.10, 0.15) |
Algorithm 2: Working of IFWA operator to select optimal route

Input: A decision matrix having \( r \) roots and \( c \) number of criteria.
\[
\begin{align*}
D &= \{ \text{root}_1, \text{root}_2, \ldots, \text{root}_r \} \\
C &= \{ c_1, c_2, \ldots, c_c \}
\end{align*}
\]

Input: Enter set of weight vectors
\[
W = \{ w_1, w_2, \ldots, w_c \}
\]

1. \( r = |D| \)
2. \( c = |C| \)
3. For \( i \leftarrow 1 \text{ to } r \) do
   4. \( \mu \leftarrow 1 \)
   5. \( \nu \leftarrow 1 \)
   6. For \( j \leftarrow 1 \text{ to } \frac{c}{2} \) do
      7. \( \mu \leftarrow \mu \times 1 - (1 - D(i, 1 : 2 : \text{end}))^{w_j} \)
      8. \( \nu \leftarrow \nu \times D(i, 2 : 2 : \text{end})^{w_j} \)
   9. Score = \( \mu - \nu \)
10. Return Score

The algorithm related to the source grouping of the input nodes (Algorithm 3) is made explicitly for making a group of the input nodes for all of the MAs, networks, and practices, using \( G \) from Algorithm 1 as an input. It determines whether the node has two or more out-degree connections and at least one connection or more in-degree connections. This is a shared node as a result. Because of its multiple routes and status as a generating group construction piece, it plays a significant part in the network. It also serves as the primary hub for linking together the nodes in the DAG network. By figuring out the roots and leaves of the network, the specific algorithm learns and determines a set of various routes that cover all the related nodes and paths. Then, it determines every path between the leaves and roots. The source grouping method evaluates all routes to determine which has the fewest nodes based on Algorithm 2, which employs fuzzy aggregation operators, in order to determine the least route. A shared node is one that is a part of numerous routes, according to our definition. This indicates that many routes can be used to access common nodes. Following is how shared nodes between various routes will be distributed: (1) A collection of nodes that are exclusive to a given route are initially assigned to each route (these nodes are referred to as private nodes on the route) and (2) the group that currently has the fewest nodes among the associated routes will be given a shared node. Each created group specifies a collection of nodes from which the sent MAs can gather information.

Algorithm 3: Source grouping of sensor nodes

Input: DAG \( G \)
Output: List of routes, Groups of source nodes

1. If in-degree connection of node \( > = 1 \) and out-degree connection of node \( > = 2 \) then
   2. Shared Node in \( G \)
3. For each root in Roots in DAG do
   4. For each leaf in Leaves in DAG do
      5. For each route in nx.all simple paths (Dag, root, leaf) do
         6. Find all routes in \( G \)
3. For each shared node routes list in DAG do
   7. If route \( > = 2 \) then
      8. Select the route with the least number of nodes
   9. For each selected least route do
      10. Add all least route nodes into a group.
11. Return list of routes, Groups of source nodes
Figure 3 is an illustration of how the algorithms’ underlying ideas work. It displays a series of routes (in this example, routes 1, 2, and 3) that cover all nodes in the network, including shared and private nodes. Private nodes are those that are exclusive to a given route. Source nodes on several routes are referred to as shared nodes. In this example, there is just one shared node, and it is present on both routes 1 and 2. A group is also a collection of assigned shared nodes and private nodes in a specific route. The groups are created by assigning shared nodes to the group of a route that contains the fewest nodes overall. As an illustration, see Figure 3. We assign the shared node to the group for route 2 since there are only two private nodes in the group for route 2, compared to the three private nodes in the group for route 1. Keep in mind that, in practice, the source sink (such as the dispatcher) and the sink node (such as the destination) may be the same (sink) node. Here, since the entire network is modelled as a DAG, it is effectively split into two nodes that each utilise a portion of the network’s links to source nodes.

Mobile agents are dispatched to groups by Algorithm 4. The first inputs are a DAG (G), a list of routes (R) (r), and a set of source nodes (gs), which are provided using Algorithm 3. T will then be initialised with a blank bundle of MA data. In order to avoid having two agents take the same group, it starts by sending mobile agents to a certain group in gs. During the journey, each MA visits nodes on the specified group gs. First, it determines whether the node is accessible and online; if so, it gathers the information and marks the node as visited. Then, it checks if the node is offline; if so, it will be marked as dead, and the agent will move to the next node in the group. If the node’s flag visited is set to true, it then moves on to visit the next node in the path. If not, the MA collects data up until its d data load and sets this node’s visited flag to true. The MA completes the tasks assigned to it, then leaves by either travelling to every online node along the predetermined route or by gathering d data on the trip. The d data load threshold for each agent ensures that the agent buffer will not become overloaded with data in a single trip.

**Algorithm 4: 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 node is online then
5. if n has been visited by any other mobile agent then
6. Mark as visited and move to the next node in the group
7. else
8. Collect data, mark as visited and move to the next node in the group
9. else
10. Node is offline
11. Mark as dead node and move to the next node in the group
12. Repeat Step 3, until all nodes in r have been visited
13. if all nodes in r have been visited then
14. MA completes the allocated tasks
15. else
16. Dispatch MA to visit node n
17. Collect data T' (not exceeding limitation d in total) from node n
18. Add data in T' to T
19. Set visited flag of node n to true
20. if T contains d data then
21. MA completes the allocated tasks
22. Return T
4. Experiments, Evaluation and Analysis of Results

4.1. Simulation Setup

Using python-based simulation, we implemented and evaluated the proposed GDMIP approach as well as current approaches from the literature, including the SMIP [15] approach, GIGM-MIP [16] approach, and CL-MIP [17]. The network model is taken from [17], where 100 sensor nodes are distributed evenly over a space, and a sink node is positioned in the middle. All of the mobile agent parameters used in the simulation are listed in Table 2.

4.2. Simulation Parameters

The sensor nodes are static and evenly distributed over a network with a size of 1000 by 500 m. Where 100 predetermined number of sensor nodes, and they are dispersed randomly throughout the network. The sink node, which is in the middle of the network, has the greater processing power and an endless supply of energy. The starting energy is the same for all of the sensor nodes. The raw data size is 2048 bits, and the radio transmission range is 60 m. The raw data reduction ratio is 0.8, the mobile agent code size is 1024 bits, the data processing rate is 50 Mbps, and the mobile agent accessing latency is 10 ms. The simulations of the trials take into account all sorts of energy usage, including transmission, idling, and sensing. All of the mobile agent parameters used in the simulation are listed in Table 2.

| Network Size | 1000 m × 500 m |
| Sensor Nodes | 100 |
| Data Size (raw) | 2048 bits |
| Code Size (MA) | 1024 bits |
| Data Processing Rate | 50 Mbps |
| Data Reduction Ratio (raw) | 0.8 |
| Aggregated Ratio | 0.9 |
| Transmission Range | 60 m |
| MA delay accessing | 10 ms |

4.3. Evaluation and Analysis of Results

The following criteria were used to assess how well various approaches performed: Task Duration and Energy Efficiency are two performance measures that are taken into account. The two performance metrics are defined in Table 3.

| Performance Metrics | Definition |
|---------------------|------------|
| Energy Consumption  | refers to the energy used by mobile agents to send and receive messages from all sensor nodes. |
| Task Duration       | refers to the period of time, on average, from when the sink sends out the mobile agents to when the last one returns to the sink. |

Energy Efficiency: The proposed GDMIP approach clearly outperforms SMIP, GIGM-MIP, and CL-MIP approaches in terms of energy utilization, as illustrated in Figure 4. In all four strategies, more energy is needed to execute tasks with more agents. However, it can be seen that the proposed GDMIP saves more energy than other methods. When the number of nodes is reduced from 100 to 10, the proposed GDMIP approach achieves a
32.3% and 14.3% energy reduction compared to SMIP 37.3% and 16.2%, GIGM-MIP 46.4% and 18.2%, and CL-MIP 47.5% and 26.6%. The CL-MIP algorithm uses the most energy. This is due to an increase in the number of mobile agent hops caused by the spread of many mobile agents throughout the sensor network. When the quantity of sensor nodes rises, the GIGM-MIP algorithm uses less energy than the CL-MIP. This is because the GIGM-MIP algorithm divides up the data collected across mobile agents in each network zone. When compared to SMIP, GIGM-MIP, and CL-MIP, the suggested GDMIP consumes less energy. This accomplishment was made possible by effective partitioning, or the grouping approach, as well as the mobile agents’ minimum travel distances inside each group. Fewer trips are required for the mobile agents in each group due to the effective segmentation of the network constructions.

Figure 4. The Impact of Number of Sensor Nodes on Energy Consumption.

Task Duration: In terms of task duration, Figure 5 compares the four methods, and it has been found that the proposed GDMIP approach performs better than the three currently used approaches, SMIP, GIGM-MIP, and CL-MIP. The proposed GDMIP delivers the best task duration of all techniques, as can be shown from the observations. The proposed GDMIP reduces task duration by 51.8% and 21.4%, compared to SMIP 55.7% and 29.7%, GIGM-MIP 57.3% and 33.5%, and CL-MIP 66.9% and 41.2%, which has the biggest delay when the number of nodes falls from 100 to 10. Each mobile agent is scheduled to visit all sensor nodes according to the defined routes chosen by the sink node in the SMIP, GIGM-MIP, and CL-MIP algorithms in order to gather data from sensor nodes. There are significant delays as a result of the increased number of mobile agent hops. The proposed GDMIP, however, offers shorter routes and better task duration because fewer nodes must be visited by the mobile agents.

Figure 5. The Impact of the Number of Sensor Nodes on Task Duration.
4.4. Varying the number of Dispatched MAs

Figures 6 and 7 show the results of the energy consumption and task duration when the number of dispatched MAs is 20.

Energy Efficiency: As can be seen in Figure 6, which illustrates how the density of the dispatched MAs affects energy usage, it is evident that the suggested GDMIP approach uses 20 MAs less than SMIP, GIGM-MIP, and CL-MIP approaches, while outperforming them in terms of energy consumption. It is evident that the suggested GDMIP saves more energy than alternative methods. When the number of nodes is reduced from 100 to 10, the suggested GDMIP approach delivers energy reductions of 42.4% and 26.2% compared to SMIP 51.6% and 33.3%, GIGM-MIP 58.5% and 35.2%, and CL-MIP 61.6 and 41.8%. The reduction in the number of hops for each MA within the groups is what allows the GDMIP technique to reduce energy consumption.

![Figure 6. The Impact of Number of Dispatched MAs on Energy Consumption.](image)

Task Duration: as can be seen in Figure 7, which evaluates the four methods in terms of task duration, this experiment dispatches only 20 MAs. It is evident that the GDMIP method was a better performer compared to the other approaches that included not only SMIP but also GIGM-MIP and CL-MIP. It is evident that the GDMIP achieved the optimal task duration compared to the other methods. GDMIP achieved 51.3% and 27.5% task duration reductions in contrast to SMIP 57.6% and 35.8%, GIGM-MIP 64.3% and 38.4%, and CL-MIP 68.3% and 45.4%, having the highest delay when the number of nodes dropped from 100 to 10. This improvement in the GDMIP approach is because a short itinerary is built for each MA in each group.

![Figure 7. The Impact of the Number of Dispatched MAs on Task Duration.](image)
5. Conclusions and Future Work

The multiple MAs approach enhances the data collection capacity of WSNs. The main aim of this research is to develop a dynamic multi-mobile agent itinerary planning approach called GDMIP, where fault tolerance, such as node failure as a result of energy depletion, is taken into consideration. The proposed GDMIP applies an intuitionistic fuzzy set using four criteria, e.g., grouping, energy efficiency, and least number of nodes, to select the shortest route for each dispatched MA. The advantages of the proposed GDMIP include energy efficiency, requiring less delay, and avoiding faulty nodes in the network. The simulation has been carried out to evaluate the performance of the proposed GDMIP. The results indicate that GDMIP has achieved significant improvements in terms of energy consumption and task delay.

There are several interesting directions for future work. We plan to include security by design to protect the GDMIP approach. The security part will focus on selecting a lightweight encryption technique to encrypt the data collected by each mobile agent and protect the overall itinerary planning from any attack. We also plan to make the shared node a leader node, which manages a set of nodes and will be responsible for receiving data from these nodes. Consequently, an MA will be dispatched to collect data from the leader node rather than all of the nodes in each group. This would result in a significant reduction in task duration. The proposed work can be extended to the fuzzy integral-based control of a robotic head for human–robot interaction.

Author Contributions: The work described in this article is a collaborative effort from all of the authors. Conceptualisation: T.A., R.H., H.A.-A. and H.M.A.F.; Methodology, S.I., M.R. and H.M.S.; Design, implementation, and generation of results, T.A. and M.H.; Analysis and interpretation of results, H.A.-A. and T.A.; Preparing the draft, review, and editing, T.A., R.H., H.A.-A. and H.M.A.F.; visualization, T.A., R.H., H.A.-A. and H.M.A.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alsbou’i, T.; Abuarqoub, A.; Hammoudeh, M.; Bandar, Z.; Nisbet, A. Information Extraction from Wireless Sensor Networks: System and Approaches. Sens. Transducers 2012, 14, 1–17.
2. Pottie, G.J.; Kaiser, W.J. Wireless Integrated Network Sensors. Commun. ACM 2000, 43, 51–58. [CrossRef]
3. Wu, C.; Rao, N.; Barhen, J.; Iyengar, S.; Vaishnavi, V.; Qi, H.; Chakrabarty, K. On computing mobile agent routes for data fusion in distributed sensor networks. IEEE Trans. Knowl. Data Eng. 2004, 16, 740–753. doi: [CrossRef]
4. Chen, G.; Wu, S.; Zhou, J.; Tung, A.K.H. Automatic Itinerary Planning for Traveling Services. IEEE Trans. Knowl. Data Eng. 2014, 26, 514–527. [CrossRef]
5. Kovács, S.; Vincze, D.; Gácsi, M.; Korondi, P. Interpolation based fuzzy automaton for human-robot interaction. IFAC Proc. Vol. 2009, 42, 317–322. [CrossRef]
6. Yoo, B.S.; Kim, J.H. Fuzzy integral-based gaze control of a robotic head for human robot interaction. IEEE Trans. Cybern. 2014, 45, 1769–1783.
7. Kuo, I.H.; Jayawardena, C. Application of fuzzy techniques in human-robot interaction-A review. In International Conference on Social Robotics; Springer: Cham, Switzerland, 2014; pp. 249–255.
8. Alsbo’i, T.; Alrifae, M.; Etywi, R.; Abdul Jawad, M. Mobile Agent Itinerary Planning Approaches in Wireless Sensor Networks-State of the Art and Current Challenges. In International Conference on Industrial Networks and Intelligent Systems; Springer: Cham, Switzerland, 2017, Volume 188, pp. 143–153. [CrossRef]
9. Gupta, G.; Misra, M.; Garg, K. Energy and Trust Aware Mobile Agent Migration Protocol for Data Aggregation in Wireless Sensor Networks. J. Netw. Comput. Appl. 2014, 41, 300–311. [CrossRef]
10. Wang, X.; Xu X.; Zeng, G.; Xu, X. A Dynamic Building Method of Mobile Agent Path Based on Referral Networks. Appl. Math. Inf. Sci. 2014, 19, 1–5. [CrossRef]
11. Mercadal, E.; Vidueira, C.; Sreenan, C.J.; Borrell, J. Improving the Dynamism of Mobile Agent Applications in Wireless Sensor Networks through Separate Itineraries. *Comput. Commun.* 2013, 36, 1011–1023. [CrossRef]

12. Camponogara, E.; Shima, R.B. Mobile Agent Routing with Time Constraints: A Resource Constrained Longest-Path Approach. *J. Univers. Comput. Sci.* 2010, 16, 372–401.

13. Rech, L.; de Oliveira, R.S.; Montez, C. Dynamic determination of the itinerary of mobile agents with timing constraints. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Compiegne, France, 19–22 September 2005; pp. 45–50. [CrossRef]

14. Xu, Y.; Qi, H. Mobile Agent Migration Modeling and Design for Target Tracking in Wireless Sensor Networks. *Ad Hoc Netw.* 2008, 6, 1–16. [CrossRef]

15. Qadori, H.; Zukarnain, Z.; Zurina, M.H.; Subramaniam, S. A Spawn Mobile Agent Itinerary Planning Approach for Energy-Efficient Data Gathering in Wireless Sensor Networks. *Sensors* 2017, 17, 1280. [CrossRef]

16. Aloui, I.; Kazar, O.; Kahloul, L.; Servigne, S. 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) 2015, 7, 116–122.

17. Chen, M.; González-Valenzuela, S.; Zhang, Y. Multi-Agent Itinerary Planning for Wireless Sensor Networks. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 22, pp. 584–597. [CrossRef]

18. Dong, M.; Ota, K.; Yang, L.T.; Chang, S.; Zhu, H.; Zhou, Z. Mobile agent-based energy-aware and user-centric data collection in wireless sensor networks. *Comput. Netw.* 2014, 74, 58–70. [CrossRef]

19. Gupta, G.P.; Misra, M.; Garg, K. Towards scalable and load-balanced mobile agents-based data aggregation for wireless sensor networks. *Comput. Electr. Eng.* 2017, 64, 262–276. [CrossRef]

20. Jana, C.; Garg, H.; Pal, M. Multi-attribute decision making for power Dombi operators under Pythagorean fuzzy information with MABAC method. *J. Ambient. Intell. Humaniz. Comput.* 2022. [CrossRef]

21. Jana, C.; Pal, M. A dynamical hybrid method to design decision making process based on GRA approach for multiple attributes problem. *Eng. Appl. Artif. Intell.* 2021, 100, 104203. [CrossRef]

22. Jana, C.; Pal, M.; Liu, P. Multiple attribute dynamic decision making method based on some complex aggregation functions in CQROF setting. *Comput. Appl. Math. 2022, 41, 103. [CrossRef]

23. Farid, H.M.A.; Riaz, M. Some generalized q-rung orthopair fuzzy Einstein interactive geometric aggregation operators with improved operational laws. *Int. J. Intell. Syst.* 2021, 36, 7239–7273. [CrossRef]

24. Athar Farid, H.M.; Riaz, M. Innovative q-Rung Orthopair Fuzzy Prioritized Interactive Aggregation Operators to Evaluate Efficient Autonomous Vehicles for Freight Transportation. *Sci. Iran.* 2022. [CrossRef]

25. Riaz, M.; Farid, H.M.A.; Shakeel, H.M.; Aslam, M.; Mohamed, S.H. Innovative q-rung orthopair fuzzy prioritized aggregation operators based on priority degrees with application to sustainable energy planning: A case study of Gwadar. *AIMS Math.* 2021, 6, 12795–12831. [CrossRef]

26. Senapati, T.; Chen, G.; Yager, R.R. Aczel–Alsina aggregation operators and their application to intuitionistic fuzzy multiple attribute decision making. *Int. J. Intell. Syst.* 2022, 37, 1529–1551. [CrossRef]

27. Senapati, T.; Chen, G.; Mesiar, R.; Yager, R.R. Intuitionistic fuzzy geometric aggregation operators in the framework of Aczel-Alsina triangular norms and their application to multiple attribute decision making. *Expert Syst. Appl.* 2022, 212, 118832. [CrossRef]

28. Senapati, T.; Yager, R.R.; Chen, G. Cubic intuitionistic WASPAS technique and its application in multi-criteria decision-making. *J. Ambient. Intell. Humaniz. Comput.* 2021, 12, 8823–8833. [CrossRef]

29. Ashraf, S.; Abdullah, S.; Mahmood, T. Aggregation operators of cubic picture fuzzy quantities and their application in decision support systems. *Korean J. Math.* 2020, 28, 343–359.

30. Ashraf, S.; Abdullah, S.; Aslam, M. Symmetric sum based aggregation operators for spherical fuzzy information: Application in multi-attribute group decision making problem. *J. Intell. Fuzzy Syst. 2020, 38, 5241–5255. [CrossRef]

31. Ali, Z.; Mahmood, T.; Aslam, M.; Chinram, R. Another view of complex intuitionistic fuzzy soft sets based on prioritized aggregation operators and their applications to multiattribute decision making. *Mathematics 2021, 9, 1922. [CrossRef]

32. Feng, F.; Zheng, Y.; Sun, B.; Akram, M. Novel score functions of generalized orthopair fuzzy membership grades with application to multiple attribute decision making. *Granul. Comput.* 2022, 7, 95–111. [CrossRef]

33. Alcantud, J.C.R.; Santos-García, G.; Akram, M. OWA aggregation operators and multi-agent decisions with Nsoft sets. *Expert Syst. Appl.* 2022, 203, 117430. [CrossRef]

34. Deveci, M.; Pamucar, D.; Gokasar, I.; Pedrycz, W.; Wen, X. Autonomous Bus Operation Alternatives in Urban Areas Using Fuzzy Dombi-Bonferroni Operator Based Decision Making Model. *IEEE Trans. Intell. Transp. Syst.* 2022. [CrossRef]

35. Qadori, H.Q.; Zukarnain, Z.A.; Hanapi, Z.M.; Subramaniam, S. FuMAM: Fuzzy-Based Mobile Agent Migration Approach for Data Gathering in Wireless Sensor Networks. *IEEE Access* 2018, 6, 15643–15652. doi: [CrossRef]

36. Tseng, Y.C.; Kuo, S.P.; Lee, H.W.; Huang, C.F. Location Tracking in a Wireless Sensor Network by Mobile Agents and Its Data Fusion Strategies. In *Information Processing in Sensor Networks*; Zhao, F., Guibas, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 625–641.

37. Chen, M.; Kwon, T.; Yuan, Y.; Choi, Y.; Leung, V.C. Mobile Agent-Based Directed Diffusion in Wireless Sensor Networks. *EURASIP J. Adv. Signal Process.* 2006, 2007, 036871. [CrossRef]
38. Zadeh, L. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
39. Atanassov, K.T. Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **1986**, *20*, 87–96. [CrossRef]
40. Xu, Z. Intuitionistic Fuzzy Aggregation Operators. *IEEE Trans. Fuzzy Syst.* **2007**, *15*, 1179–1187. [CrossRef]