Indirect spatio-temporal communication for SMWSN-based collaborative data-foraging in dynamic environments

por

Fabiola Marcos Solis

Tesis sometida como requisito parcial para obtener el grado de

MAESTRA EN CIENCIAS EN EL ÁREA DE CIENCIAS COMPUTACIONALES

Instituto Nacional de Astrofísica, Óptica y Electrónica
Febrero 2018
Tonantzintla, Puebla

Supervisada por:

Dr. Saúl Eduardo Pomares Hernández, INAOE
Dra. Lil María Xibai Rodríguez Henríquez, CONACYT-INAOE
Dr. José Roberto Pérez Cruz, CONACYT-UMSNH

Dedication

To my mother,

whose strong spirit sustains me

Acknowledgements

“We are what we think.
All that we are arises with out thoughts.
With our thoughts we make the world"
— Buddha

It is a pleasure to acknowledge my family and friends. Thank you for your support and encouragement.

I would like to thank Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) for the opportunity to be one of its students.

Special thanks to my advisors, who shared their opinions and gave me much useful advice.

This thesis work was support by the Consejo Nacional de Ciencia y Tecnología (CONACYT) through the Master scholarship number 422280.

Abstract

In a Sparse Mobile Wireless Sensor Network (SMWSN) the mobile sensors, called Mobile Data Collectors (MDCs), are spread over a large area to retrieve environmental data. Due to the dynamic movement of the MDCs, they should not be restricted to share the same area at the same time. It is sometimes impossible to say that one of two samples occurred first, especially if the samples were obtained by different MDCs due to the lack of perfectly synchronized clocks. In this kind of conditions, to exchange information in a direct manner is not possible without an enduring transmission link. However, for some applications, it is required for the MDCs to collaborate and exchange information about constantly changing features. To achieve a profitable collaboration among MDCs, an essential task is to determine the context of the retrieved data; which requires relating the spatial and temporal domains in a feasible way. We propose an indirect spatio-temporal communication among a group of MDCs oriented to retrieve data applying a collaborative approach for dynamic environments. This approach is inspired from stigmergy to accomplish an indirect communication among MDCs to exchange information. The indirect spatio-temporal communication is performed through spatial-temporal relations based on causal and fuzzy causal dependencies.

Resumen

En una Red de Sensores Móviles Inalámbricos Escasa (SMWSN, por sus siglas en inglés) los sensores móviles, llamados recolectores móviles de datos (MDCs), son distribuidos sobre una gran área para recuperar datos del ambiente. Debido al movimiento dinámico de los MDCs, no deberían ser restringidos a compartir la misma área al mismo tiempo. Algunas veces es imposible decir si una de dos muestras ocurre primero, especialmente si las muestras fueron obtenidas por diferentes MDCs debido a la escasez de relojes perfectamente sincronizados. En este tipo de condiciones, intercambiar información de manera directa no es posible sin un enlace de transmisión perdurable. Sin embargo, para algunas aplicaciones se requiere que los MDCs colaboren e intercambien información acerca de características cambiantes constantemente. Para lograr una colaboración entre los MDCs, una tarea esencial es determinar el contexto del dato recuperado; el cual requiere relacionar los dominios espaciales y temporales de manera factible. Proponemos una comunicación indirecta espacio-temporal entre un grupo de MDCs orientado a la recuperación de datos con un enfoque colaborativo para ambientes dinámicos. Este enfoque está inspirado en la estigmergia para lograr una comunicación indirecta entre los MDCs para intercambiar información. La comunicación espacio-temporal es desarrollada a través de relaciones espacio-temporales basadas en dependencias causales y causales difusas.

Contents

Notation Table

bs	$\triangleq$	Base station
MDF	$\triangleq$	Set of Mobile Data Foragers where each $mdf \in MDF = \{mdf_1, mdf_2, \cdots\}$
OA_h	$\triangleq$	Operational Area to model the environment, with degree h
$r_1, r_2, \cdots, r_u$	$\triangleq$	Set of regions that conform OA
SOA	$\triangleq$	Set of sub-operational areas where each $S_{oa} \in SOA = \{S_{oa_1}, S_{oa_2}, \cdots, S_{oa_{12}}\}$
$steps$	$\triangleq$	Number of steps required to traverse an S_{oa}
CD	$\triangleq$	Number of steps between the timestamp and $steps$
$tsg_{f_{(u,v)}}$	$\triangleq$	Timestamp in the global timeline of a pheromone
at	$\triangleq$	Number of steps to arrive to bs according the global timeline
old_f	$\triangleq$	Antiqueness of a pheromone f
R_{insp}	$\triangleq$	Set of regions of inspection where $R_{insp} \subseteq OA$
ROI	$\triangleq$	Set of regions of interest where $ROI \subseteq R_{insp}$
$Phero$	$\triangleq$	Set of pheromones where each $f \in PHERO$
IPR	$\triangleq$	Intensity per region of interest $roi \in ROI$

Chapter 1

Introduction

1.1 Motivation

A Wireless Sensor Network (WSN) consists of hundreds or thousands of tiny devices with a limited battery which measures and collects data from a sensing area, and transfers it to a base station or *sink*[1] through wireless communication [Di Francesco et al., 2011]. In a traditional WSN architecture, the network is assumed as dense where the sensors are static and two nodes can communicate with each other through multihop paths . As a consequence, the nodes closer to the sink are overloaded when compared with the others, and subject to premature depletion. However, mobility has been introduced in order to increase the capabilities of the WSN.

A Mobile Wireless Sensor Network (MWSN) is defined as a special and versatile kind of WSN, in which one or more nodes are mobile [Sayyed and Becker, 2015] and interact with the physical environment [Yick et al., 2008]. Several MWSN architectures have been proposed to retrieve environmental data. In some of these architectures, a few mobile sensors are deployed over a large geographical area and move reaching isolated regions in it. In an ideal context, the communication among these sensors can be achieved directly through enduring transmission links. Nevertheless, since the transmission range of the nodes is smaller than the distance between neighboring nodes, the communication among them cannot always

[1]The destination or consumer of messages originated by sensors [Sayyed and Becker, 2015].

be achieved. This kind of networks are known as Sparse Mobile Wireless Sensor Network (SMWSN).

Some applications of SMWSN are the monitoring of environmental conditions (e.g., underwater monitoring) [Vasilescu et al., 2005], target tracking (e.g., wildlife tracking, surveillance, etc.) [Juang et al., 2002][Qu et al., 2015] and healthcare applications [Yan et al., 2010]. Figure 1.1 depicts an example of SMWSN architectures based on mobile nodes for the monitoring of a sensing area. Figure 1.1a shows a series of static sensors that were deployed over the environment. The static sensors are communicate in a multi-hop fashion. Figure 1.1b depicts an scenario with stationary and mobile nodes. The mobile nodes know the location of the sensors, hence, they have a trajectory and directly collect the data from them. Figure 1.1c depicts an scenario that take advantage of the static and mobile nodes. In this case, the static nodes are capable to communicate with the mobile node in near range. Mobile nodes can form an ad hoc network by communication with each other. They are responsable to pick up the data and forwarding to base station or access points. In both architectures, Figure 1.1b and Figure 1.1c, the mobile nodes move to anywhere at anytime if needed. This makes it difficult for the nodes to have direct communication permanently. Thus, they would not share the information about the environment.

1.2 Problem description

In a Sparse Mobile Wireless Sensor Network (SMWSN) the mobile nodes, called Mobile Data Collectors (MDCs), start with an initial deployment and then they disperse in a sampling area in order to collect information. Despite the scarcity of direct enduring transmission links, for some applications, it is required that the MDCs exchange information about environmental conditions and perform profitable tasks. An example of such applications is the collaborative data-foraging (Figure 1.2), which is oriented to periodically searching valuable data sources over

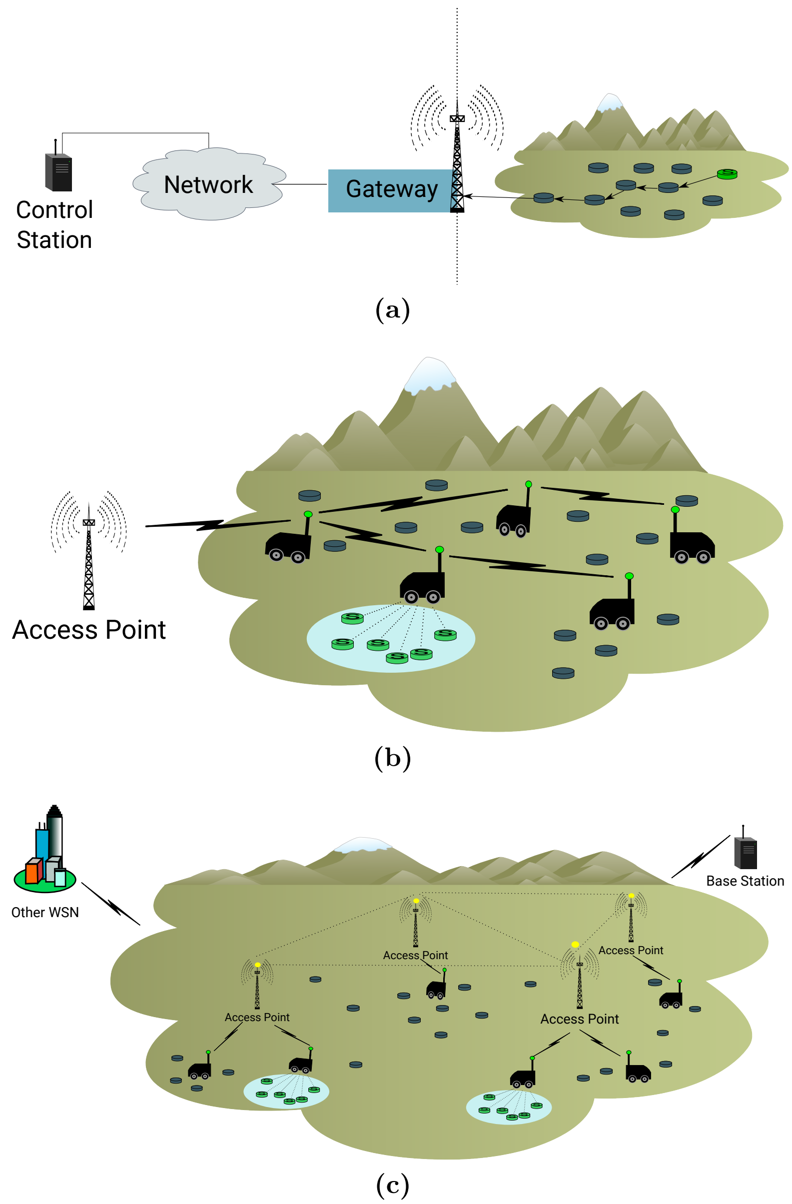


Figure 1.1: Graphical representations of different architectures of Sparse Wireless Sensor Network based on mobile elements. a) *Planar*: data is routed from the source sensor to the access point in a multi-hop fashion. b) *Two-tiered*: a set of mobile nodes construct an overlay network, carrying data and establishing connectivity among fixed nodes. c) *Three-tiered*: the static nodes pass the retrieved data to the mobile nodes, which then forward data to the access points where finally data is uploaded to a centralized data server. Figure adopted from [Munir et al., 2007, p. 2 - 3]

a particular area. Besides, in environments where the location and relevance of the data are continually changing, not all the area provides useful information. Thus, it is necessary that data gathering is performed selectively, similarly as some animals search for valuable wild food resources [Kramer, 2001]. For this kind of applications, an essential task is to determine the context of the retrieved data through the collaboration among MDCs. In environments where data sources may change or disappear, the data context is related to the location of the sources and when they were sampled. Therefore, the collaboration among MDCs must be oriented to exchange spatio-temporal information about the environment in a feasible way. Due to the distributed nature of this problem, it is sometimes impossible to say that one of two samples occurred first, especially if that were met by different MDCs. This is because an instantaneous observation across various locations is not possible due to the lack of perfectly synchronized clocks [Chandra and Kshemkalyani, 2005].

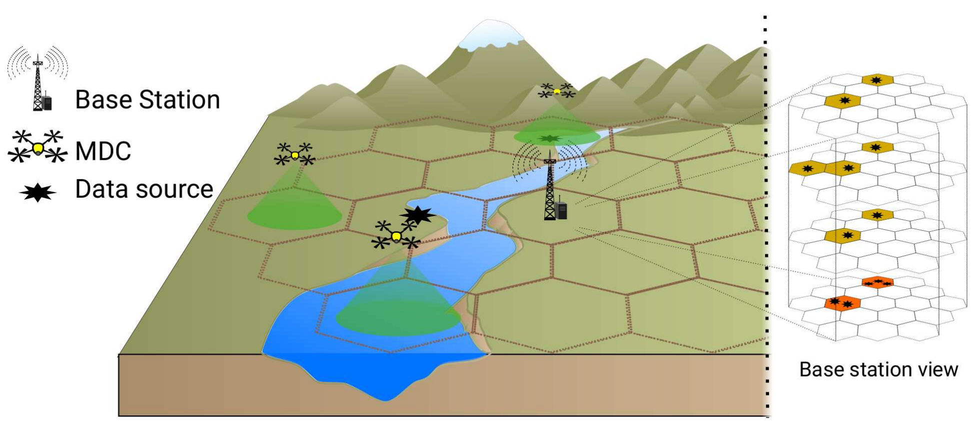


Figure 1.2: Collaborative data-foraging. MDC (represented by aerial vehicles with four rotors) performs sensing task in order to find profitable data sources (black stars). Once data sources are found, the MDC returns to the base station (represented as an antenna) to download the collected data. Finally, the base station assembles the collected data and shares the location of the data sources (depicted as overlapped sets of hexagons) to later perform the harvesting.

1.3 Proposed solution

In this work we propose an indirect spatio-temporal communication among a group of MDCs, oriented to retrieve data with a collaborative approach for dynamic environments. To tackle the lack of enduring transmission links in an SMWSN, we take inspiration from stigmergy[2] to achieve indirect communication among various MDCs by using artificial pheromones. To determine, in a distributed fashion, the spatio-temporal context of the retrieved data without requiring global references, such as perfectly synchronized clocks; we propose to perform the pheromone-based communication through causal and fuzzy causal protocols. According to the Fuzzy-Causal Relation [Perez Cruz and Pomares Hernandez, 2014], by relating the logical/temporal domain (determined by the causal order) with the spatial domain, a degree of "*closeness*" between events can be inferred and it can be established "*how long ago*" an event happened before another.

1.4 Research hypothesis and objectives

1.4.1 Research hypothesis

Indirect spatio-temporal group communication for dynamic environments, oriented towards collaborative data foraging, can be performed through causal and fuzzy causal dependencies.

1.4.2 Main objective

To design and develop an indirect spatial-temporal communication mechanism oriented to collaborative data-foraging satisfying the features and constraints of a Sparse Mobile Wireless Sensor Network (SMWSN).

[2] "The stigmergy is a method of indirect communication using signals through physical media which trigger responses among the insects" [Dipple et al., 2014].

1.4.3 Specific objectives

1. To design an indirect communication model among Mobile Data Collectors (MDCs) based on the stigmergy principle of pheromone secretion.

2. To develop a collaborative group communication protocol based on the indirect communication model.

3. To design and develop a reconnaissance mechanism for the identification of Regions of Interest (ROI) over a particular area for a single MDC.

4. To develop a data harvesting mechanism to retrieve profitable data, considering the ROI identified by each MDC, through the collaborative group communication protocol.

1.5 Methodology

To achieve the stated objectives, we propose the following summarized methodology:

- Definition of the pheromone as an Abstract Data Type (ADT), by modeling its inherent operations: creation, aggregation and decay; through causal and fuzzy causal relations.

- Development of an indirect communication model based on the secretion of pheromones as a mechanism to exchange spatio-temporal information.

- Development of a group communication protocol based on the previous indirect spatio-temporal communication model.

- Design and development of a functional model to identify ROI based on the intensity of the pheromone.

- Development of a collaborative and adaptive reconnaissance mechanism to control the mobility of a single MDC to identify ROI.
- Design and development of an extension of the reconnaissance mechanism to support collaborative interactions among various MDCs, based on the indirect spatio-temporal group communication protocol.
- Design and development of a data harvesting mechanism that considers the ROI identified by the reconnaissance mechanism to perform data oversampling through various MDCs.
- Analysis of the proposed protocol to determine the bounds of the time and resources' requirements (storage, communication and computational overheads).
- Simulation of the data harvesting mechanism to determine if the resources and temporal constraints are accomplished and whether the mechanism is adaptable to the environmental dynamism.

1.6 Document organization

The organization of this document is as follows. In Chapter 2 the main concepts about communication, causal ordering and fuzzy sets are presented. Chapter 3 discusses the related work associated with the data collection and the communication protocols used for non-enduring transmission links. In addition, a taxonomy is proposed to organize it. An indirect spatio-temporal communication inspired in the biological principle of stigmergy is described in Chapter 4. Chapter 5 describes the implementation of the proposed protocol, presented in Chapter 4, in the mechanism of collaborative data-foraging. Besides, the results obtained from a series of experiments are reported. Finally, Chapter 6 shows the conclusion and future work for this research.

Chapter 2

The fundamentals

In this Chapter, the principal definitions used in this document are described. First, given the distributed nature of the problem, the concepts related to the distributed systems, its elements and communication paradigms used in this kind of systems are defined. Later, some concepts of coordination and communication borrowed from nature, e.g. stigmergy and foraging are discussed. Finally, the causal dependencies' definitions used to model the coordination and behavior of natural mechanisms within a distributed system are introduced.

2.1 Distributed systems

A distributed system is composed by different entities spatially separated, which communicate with each other by exchanging messages [Coulouris et al., 2012] [Lamport, 1978].

A typical distributed system is described by a model composed by the following elements:

- **Processes**. Programs or instances of programs running simultaneously with other programs. Each process belongs to the set $P = \{p_1, p_2, \ldots\}$.

- **Messages**. Abstractions to represent frames or packets in a computer network, which can contain arbitrarily complex data structures. In a typical distributed system a process can only communicate with other processes by message passing over a communication network. Each exchanged message in the system belongs to the set M.

- **Events.** An event denotes an indivisible atomic action which occurs at processes [Mattern et al., 1989]. In a distributed system, a process can only execute two kinds of events: *internal events* and *external events*.

  An internal event affects only the process at which it occurs. An external event is an action that involves communication and indicates the information flow with another process affecting the global system state. There are two types of external events:

  1. **Send event**: refers to the emission of a message, executed by a process.
  2. **Receive event**: refers to the notification on the arrival of a message in a process.

In a distributed system each entity (host, printer, file, process, user, etc) has its own physical clock, which can be used by local processes to obtain the value of the current time. However, even if two processes read their clock at the same time, their local clock may supply different time values [Coulouris et al., 2012]. In many cases, it is important to determine whether an event (sending or receiving a message) at one process occurred before, after or concurrently with another event at another process. In the absence of a global physical time, in a distributed system it is often impossible to determine the system's causal order.

2.2 Communication in distributed systems

Communication in a distributed system is based on message passing among the entities supposing a communication channel between them. There are two paradigms for message passing communication between processes: 1) the sender knows the destination processes and 2) there is a designed fixed location for receiving a message. The first paradigm is called direct communication and it uses direct names; the latter is called indirect communication [Jia and Zhou, 2004]. Next, these paradigms are described in detail.

2.2.1 Direct communication

Direct communication represents a two-way relationship between a sender and a receiver. The sender explicitly directs messages/invocations to the associated receiver (Figure 2.1). The receiver is also generally aware of the identity of the sender, and in most cases both parties, sender and receiver, must exist at the same time [Coulouris et al., 2012].

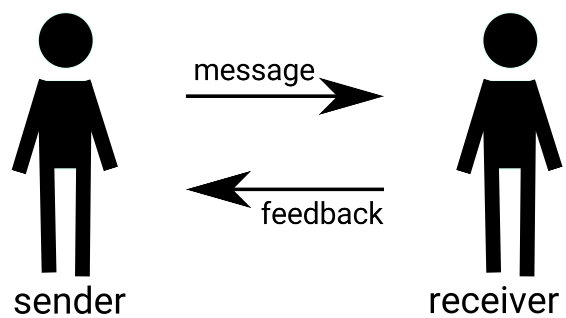


Figure 2.1: Direct communication between sender and receiver.

2.2.2 Indirect communication

Indirect communication is defined as a communication between entities, in a distributed system, through an intermediary without a direct coupling between the sender and the receiver(s) [Coulouris et al., 2012]. Figure 2.2 depicts a simple example: the sender leaves a message into a mailbox, where the receiver(s) will take the message later. In this case, the presence of the sender is not required to deliver the message and the receiver does not need to be aware of the sender. That are two key properties of indirect communication, called space uncoupling and time uncoupling which are described below.

- In *Space uncoupling* the sender does not need to know the identity of the receiver(s), and vice versa.

- *Time uncoupling* refers to the lifetime's independence of the sender and receiver(s). In other words, the sender and receiver(s) do not need to exist at the same time to communicate.

While direct communication needs a perdurable communication channel and knowledge about the receiver(s), indirect communication does not depend of that.

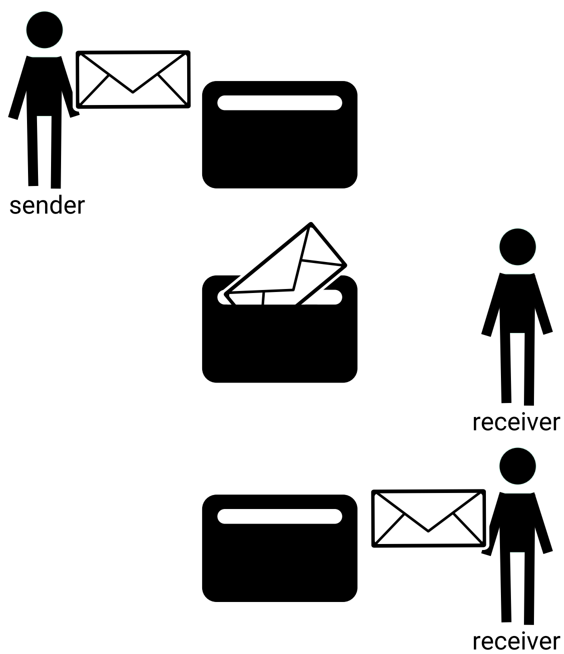


Figure 2.2: Indirect communication between sender and receiver

2.3 Coordination mechanisms in natural systems

Social insect societies are distributed systems that, in spite of the simplicity of their individuals, present a highly structured social organization. As a result of this organization, insect societies can accomplish complex tasks that in most cases exceed the individual capacities of single individuals [Dorigo et al., 2000]. Some mechanisms to coordinate and organize social insects are stigmergy and foraging. These mechanisms are described below.

2.3.1 Stigmergy

The term stigmergy was first introduced in 1959 by Pierre-Paul Grassé [Grassé, 1959] to explain how termites appear to coordinate without an obvious management structure. Grassé's research described a method of indirect communication using environment-mediated signals to trigger responses from other colony members [Dipple et al., 2014]. According to a direct translation of Grassé's research to English by Holland et al. [Holland and Melhuish, 1999], the following definition is provided:

"The stimulation of the workers by the very performances they have achieved is a significant one inducing accurate and adaptable response, and has been named stigmergy."

The societies of termite *Macrotermes* use soil pellets impregnated with a pheromone to build pillars. The accumulation of material reinforces the attractiveness of deposits through the diffusing of pheromone emitted by the pellets. First, the termites deposit pellets in a randomly manner, this stimulates the workers to accumulate more materials. When one of the deposits reaches a critical size, the coordination starts if the group of builders is sufficiently large, otherwise, the pheromone disappears.

The stigmergy phenomenon can be observed in other insect societies such as ants. In the double-bridge experiment by Goss et al. [Goss et al., 1989] with the Argentine ant *Iridomyrmex humilis*, a food source is separated from the nest by a bridge with two branches with different lengths (Figure 2.3a). Although initially, both branches have the same probability of being selected (Figure 2.3b), some minutes later ants chose a branch where explorers had laid pheromones.

More ants chose the short branch with the more significant amount of pheromone (Figure 2.3c).

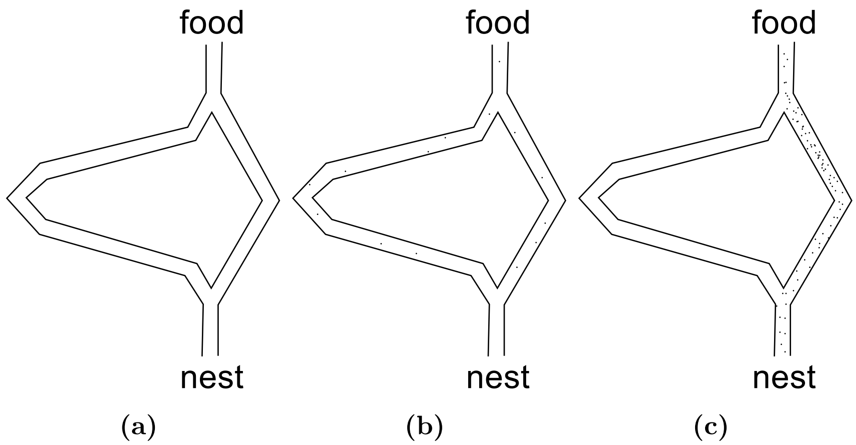


Figure 2.3: Shortcuts in the Argentine ant experiment. a) Nest and food separated by two branches. b) Both branches are chosen with the same probability. c) Ants prefer the branch with more pheromones.

2.3.2 Foraging

Foraging is the set of processes by which organisms acquire energy and nutrients, whether the food is directly consumed, stored for later consumption or given to other individual. In other words, foraging is a cyclical activity in which a series of behavioral acts leads to the final consumption of each unit of food [Kramer, 2001]. Beckers et al. [Beckers et al., 1989] proposed the following categories of foraging strategy:

- **Individual foraging**. The individual foraging is the foraging without systematic cooperation or communication in the discovery, capture or transport of prey items. Each forager leaves the nest, searches for food and transports it individually.

- **Group foraging**. In group foraging, a scout having discovered a food item returns to the nest and transmits the information concerning its location to the other foragers.

The group foraging has several advantages over individual foraging. According to Jackson et al. [Jackson and Ratnieks, 2006], the presence of many individuals can increase system reliability, and work can also be organized more efficiently through division of labor and task partitioning. In social insects' societies, the group foraging behavior is better known. Social insects live in a dynamic and competitive environment in which food sources of variable quality are constantly changing in location. In such environment, if the individuals can share information about it to the colony then its workers could reach quickly to the best food sources.

Jackson and Ratnieks [Jackson and Ratnieks, 2006] mention a couple of examples of group foraging in social insects. In honeybees, the waggle dance recruits additional foragers but also directs them to the food. However, honeybees have another dance, the vibratory signal, which helps to recruit more foragers but does not guide them to food. On the other side, for trail-following ants, the use of several trail pheromones that differ in their persistence provides memory over different time scales. In particular, a non-volatile pheromone can provide a longer-term memory, while a volatile pheromone can allow rapid choice among potential feeding location by quickly 'forgetting' depleted locations.

Natural behavior inspires to researchers in Computer Science in many scenarios. The simplicity of the models allows to researchers to propose efficient solutions. For example, the remarkable feature of stigmergy and foraging mechanisms of coordination is the no dependency of a direct communication among the individuals, which can be addressed to distributed systems communication issues. Besides, the shared information is useful not only to guide the food sources, the information coordinates the labors and activities of colony members.

2.4 Fuzzy sets

Fuzzy sets provide a mathematical way to represent vagueness and fuzziness of human factors[1]. In other words, fuzzy sets provide a natural manner of dealing with problems in which there are not clear criteria to define the membership class instead of crisp variables.

Zadeh [Zadeh, 1965] establishes that a fuzzy set is a class of objects with a continuum of grades of membership. This class is characterized by a membership function assigning to each object a grade of membership. The formal definition of a fuzzy set is described as follows:

Definition 2.1 (Zadeh, 1965)**.** *A fuzzy set A is a membership function $\mu_A(x)$ that maps the elements of a domain or universe X to the elements of the interval [0, 1]: $\mu_A : X \rightarrow [0, 1]$, representing the degree of membership of x in A. The closer the value of $\mu_A(x)$ to 1, the higher the degree of membership of x in A.*

A fuzzy set A can be represented as a set of pairs of values: each element $x \in X$ with its degree of membership in A.

$$A = (x, \mu_A(x))|x \in X)$$

Definition 2.2 (Ross, 2010)**.** *Fuzzification is the conversion of a precise quantity to a fuzzy quantity.*

Generally, the fuzzification of a real value is performed by using intuition, experience and an analysis of the set of conditions associated to the input variables. The most used fuzzifiers, are those based on triangular (Figure 2.4) and trapezoidal (Figure 2.5) functions:

[1]The consideration of human characteristics, expectations, and behaviors in the design of the things people use in their work and everyday lives and of the environments in which they work and live [Evans and Karwowski, 1986]

Definition 2.3 (Galindo, 2005). *The triangular membership function is specified by:*

$$\mu_A(x) = max[min\{\frac{x-a}{b-a}; \frac{c-x}{c-b}\}; 0]$$

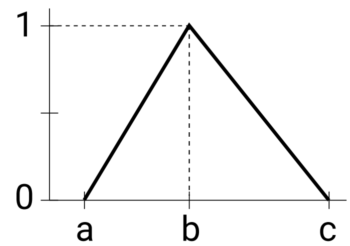


Figure 2.4: Triangular function

Where a, b and c are parameters that delimit a fuzzy set A. a corresponds to the beginning of the triangular membership function, b the input value that has the largest membership and, c the ending of the function.

Definition 2.4 (Galindo, 2005). *The trapezoidal membership function is specified by:*

$$\mu_A(x) = max[min\{\frac{x-a}{b-a}; 1; \frac{d-x}{d-c}\}; 0]$$

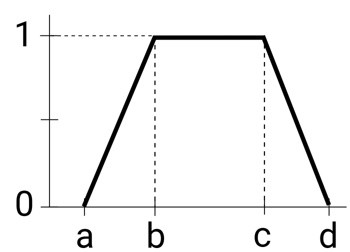


Figure 2.5: Trapezoidal function

Where a, b, c and d correspond to the beginning of the membership function (a), the boundaries of the range with a membership value equal to 1 (b and c) and the ending of the function (d).

Definition 2.5 (Lee, 1990). *Defuzzification is the conversion of a fuzzy quantity to a precise quantity.*

Defuzzification can be performed in several ways; however, the most used defuzzification methods are those based on the center of area or center of gravity.

- **Centroid of Area (COA) method**. This procedure is the most prevalent and physically appealing of all the defuzzification methods [Sugeno, 1985, Lee, 1990]; it is given by the algebraic expression:

$$COA = \frac{\int \mu_A(x) \cdot x dx}{\int \mu_A(x) dx}$$

  where $\int$ denotes an algebraic integration.

- **Weighted average (WA) method**. The weighted average method is the most frequently method used in fuzzy applications since it is one of the more computationally efficient methods [Ross, 2010]. The only restriction is that the output membership functions must be symmetrical[Ross, 2009]. It is given by the algebraic expression:

$$WA = \frac{\sum \mu_A(\overline{x}) \cdot \overline{x}}{\sum \mu_A(\overline{x})}$$

  where $\sum$ denotes the algebraic sum and $\overline{x}$ is the centroid of each symmetric output membership function. The weighted average method is calculated by weighting each membership function in the output by its respective maximum membership value.

The linguistic variables [Zadeh, 1973] are variables whose values are represented using linguistic terms (low, medium, high, very high, etc).

Definition 2.6 (Lee, 1990)**.** *A linguistic variable is characterized by the tuple* $(x, T(x), U, G, M)$

where:

- x is the names of the variable
- $T(x)$ is the set of name of linguistic values of x
- U is the universe of discourse of variable x
- G is a syntactic rule to generate linguistic terms
- M is a semantic rule that associates each value with its meaning

2.5 Fuzzy inference system

A fuzzy inference system (FIS) is a way to transform an input space in an output space, using fuzzy logic. The FIS attempts to formalize, using the fuzzy logic, human thinking and natural language. Generally, a FIS has four modules [Lee, 1990]:

- ***Fuzzification module:*** transforms the system inputs, which are crisp numbers, into memberships to fuzzy sets. This is done by applying a fuzzification function:

  $$x = \textit{fuzzifier}(x_0)$$

  where x_0 is a crisp input value and x is a fuzzy set.

- ***Knowledge base:*** stores if-then rules provided by experts.

- ***Inference engine:*** is capable of simulating the human reasoning process based on fuzzy concepts through fuzzy implication and if-then rules.

- ***Defuzzification module:*** transforms the memberships to fuzzy sets, obtained by the inference engine, into a crisp value.

  $$z_0 = \textit{defuzzifier}(z)$$

where z_0 is the crisp output value.

The most used FIS are the Mamdani type [Mamdani and Assilian, 1975] and the Sugeno type [Takagi and Sugeno, 1985].

- In the Mamdani systems the inputs and the outputs of the inference engine are fuzzy

$$\text{If } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z \text{ is } C$$

where A, B and C there are the membership functions of three fuzzy sets.

- In the Sugeno systems, the inputs of the inference engine are fuzzy and the output is "crisp"

$$\text{If } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z = f(x, y)$$

2.6 Causal dependencies

The execution of a system can be described in terms of events and their ordering is possible despite of the lack of accurate physical clocks. Lamport [Lamport, 1978] proposed a model of logical time used to provide an ordering among the events at processes running in a distributed system. Some definitions based on logical time defined by Lamport are presented as follows.

2.6.1 Happened-before relation

A causal order [Lamport, 1978] establishes a precedence relation between two events in the following way: let $e1$ and $e2$ be two events causally related, it is said that $e1$ happened before $e2$ if there is an information flow from $e1$ to $e2$, and given such a relation, $e1$ must be processed before $e2$.

Definition 2.7 (Lamport, 1978)**.** *The happened-before relation (HBR), "$\rightarrow$", is the smallest relation on a set of events E satisfying the following three conditions:*

1. *If a and b are events in the same process, and a comes before b, then "$a \rightarrow b$".*

2. *If a is the sending of a message by one process and b is the receipt of the same message by another process, then "$a \rightarrow b$".*

3. *If "$a \rightarrow b$" and "$b \rightarrow c$", then "$a \rightarrow c$".*

Besides, Lamport [Lamport, 1978] defines that a pair of events is concurrently related "$a \parallel b$" as follows:

Definition 2.8 (Lamport, 1978)**.** *Two distinct events a and b are said to be concurrent if "$a \nrightarrow b$" and " $b \nrightarrow a$".*

2.6.2 Immediate dependency relation

The Immediate Dependency Relation (IDR) [Hernandez et al., 2004] is the propagation threshold of the control information, regarding the messages sent in the causal past that must be transmitted to ensure a causal delivery, denoted by "$\downarrow$".

Definition 2.9 (Hernandez, 2004)**.** *Immediate dependency relation (IDR) is the transitive reduction of the HBR, where two events a, $b \in E$ have an immediate dependency relation if:*

$$a \downarrow b \Leftrightarrow [(a \rightarrow b) \wedge \forall c \in E, \neg(a \rightarrow c \rightarrow b)]$$

2.6.3 Causal distance

The causal distance [Dominguez et al., 2005] between two causally dependent events is the greatest number of pairwise dependent events sent between them plus one. Formally it is defined as follows:

Definition 2.10 (Dominguez, 2005)**.** *The distance $d(e, e')$ is defined for any pair of events e and $e' \in E$ such that $e \to e'$: $d(e, e')$ is the greatest integer n such that for some sequence of events $(e_i, i = 0 \ldots n)$ with $e = e_0$ and $e' = e_n$, we have $e_i \downarrow e_{i+1}$ for all $i = 0 \ldots n - 1$.*

2.6.4 Fuzzy-causal dependencies

Although the causal dependencies can be used to order events without global references, it does not give information about the time that has elapsed between a pair of events. In order to mitigate this problem, Pérez and Pomares. [Perez Cruz and Pomares Hernandez, 2014] define a fuzzy-causal relation in such a way that a *degree of closeness* among events can be inferred, considering the information about the spatial and logical/temporal distance of the events' sources.

The fuzzy-causal relation relates the logical/temporal domain with the spatial domain in the following way:

"how ancient and how far it happened imply how close an event e_1 happened before an event e_2"

To achieve this, three linguistic variables (see Definition 2.6) are defined:

- Causal distance (CD), whose universe of discourse is the logical/temporal domain;
- Physical distance (PD), whose universe of discourse is the spatial domain;
- and Fuzzy-causal closeness (FCC), whose universe of discourse is the degree of closeness among events considering both logical/temporal and spatial domains.

The fuzzy-causal relation (FCR), denoted by $\xrightarrow{\lambda}$, is formally defined as follows:

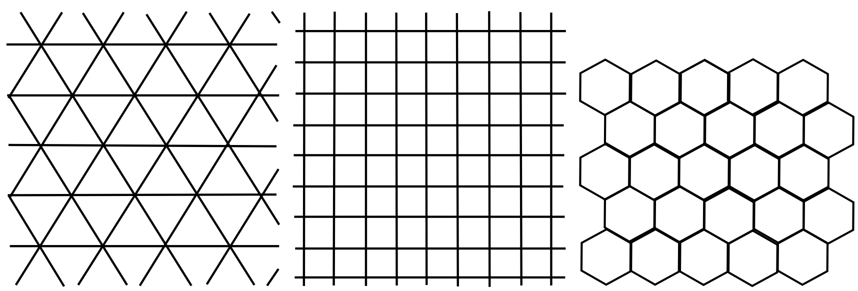
Definition 2.11 (Cruz, 2014)**.** *The FCR over a set of events must satisfy:*

1. $a \xrightarrow{\lambda} b$ *If* $a \to b$ *and* $0 < FCC < \varphi_8$

2. $a \xrightarrow{\lambda} b$ *If* $\exists c | a \xrightarrow{\lambda} c \xrightarrow{\lambda} b$ *and* $0 < FCC < \varphi_8$

2.7 Tessellation and sensors networks

A tiling of a plane is a family of sets, called tiles, that cover this plane without gaps or overlaps [Grunbaum and Shephard, 1977]. Tilings are also known as tessellations, pavings or mosaics. If the tilings are formed by regular polygons, then they are called regular and uniform tilings. There are only three regular polygons to build regular tilings with the same polygon and no gaps: equilateral triangles, squares and regular hexagons. Figure 2.6 depicts the tessellations formed by these polygons.



(a) **(b)** **(c)**

Figure 2.6: Uniform tessellations

Tessellations are used for various applications in sensor networks, specially, the square (Figure 2.6b) and the hexagonal (Figure 2.6c) grids. In the hexagonal tessellation, each hexagon has six neighbors covering the surroundings from all directions. In coverage problems [Djidjev and Potkonjak, 2012], the task of tessellation is a special coverage case where the goal is to cover a finite two-dimensional space using the repetition of a single or a finite number of geometric shapes. While in cellular networks [Nandi et al., 2014], a large number of base stations is expected to cover a communication region. Such coverage can be achieved by placing the base stations according to a regular plane tessellation.

Other application areas are the virtual infrastructures over the physical network. Such infrastructures have been investigated as an efficient strategy for data dissemination. For example, hexagonal cell-based data dissemination (HexDD) [Erman et al., 2012] is a geographical routing protocol based on this virtual infrastructure concept, proposing *rendezvous* region for events and queries. Another example is GCA [Chen and Xu, 2005] that builds a cellular-like structure assisted by the geographical location of each node. Liu et al. [Liu et al., 2006] replace the square grid with a hexagonal grid (honeycomb) to extend the lifetime to conserve energy in Wireless Sensor Networks. Sharieh et al. [Sharieh et al., 2008] introduces a topological structure Hex-Cell which requires less knowledge of the network interconnections and brings about less communication cost.

2.8 Summary

Important concepts for this thesis were presented in this Chapter. On the one hand, the natural mechanisms that inspired this work were discussed, these are related to the natural behaviors in social insects, their coordination and communication means. Despite the simplicity of individuals in those societies, they are capable of sharing environmental information through indirect communication in order to find profitable food sources. On the other hand, some concepts of distributed systems were introduced, these will allow us to model a mechanism to search and retrieve profitable data sources from dynamic environments inspired by the biological principle of stigmergy.

Chapter 3

Related Work

In Sparse Mobile Wireless Sensors Networks (SMWSN) many protocols have been proposed with the purpose to retrieve data from the environment. Unlike data collection (or data gathering), where data are collected indiscriminately, data-foraging performs collection tasks in a selective manner (performing preemptive reconnaissance and identification of regions of interest). Nevertheless, some of the solutions kindred with data collection are nearly related to data-foraging. These solutions gather data from deployed static sensors in the sampling area or select environmental data directly. Thus, different protocols have been addressed for this purpose. On the one hand, the proposed solutions to pick data of the sensors use direct or indirect communication allowing information exchange among the entities (sensor nodes and sink) through a mobile data collector (MDC). On the other hand, most of the solutions addressed to gather data directly from the environment opt by direct communication to attain a collaborative performance among the MDCs. Castanẽda et al. [Castañeda Cisneros et al., 2017] proposed a solution for data-foraging, which implements reconnaissance and data gathering. However, this solution does not consider a collaborative criterion due to use of a single MDC.

In the following Sections, the solutions proposed in the literature related to retrieving data in SMWSN are discussed. In addition, a taxonomy is presented in Figure 3.1 that summarizes the relevant work found in the literature.

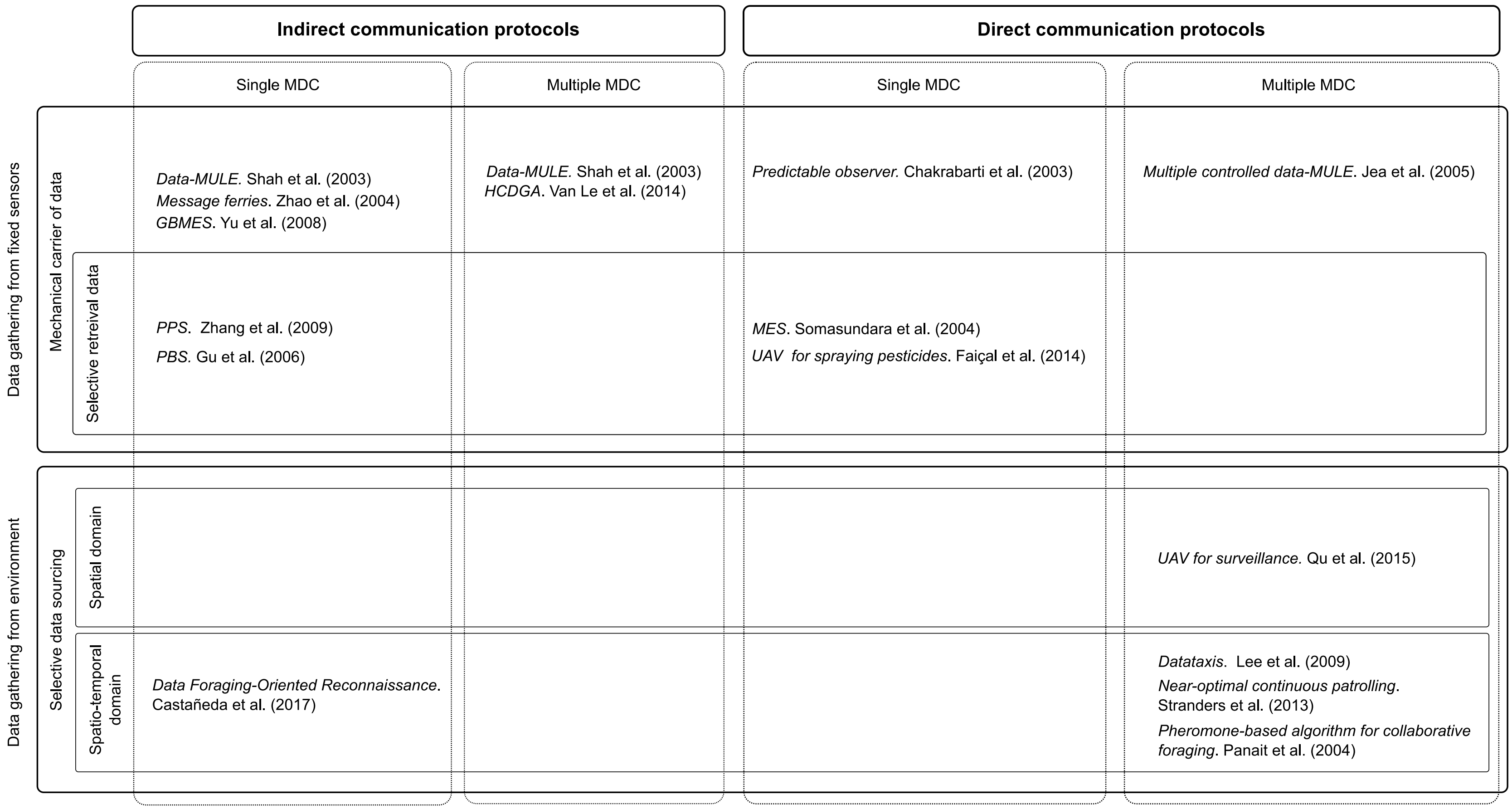


Figure 3.1: Related work taxonomy. Data gathering protocols related with data-foraging.

3.1 Data gathering from fixed sensors

The traditional Wireless Sensor Networks (WSN) architectures are based on the assumption that the network is dense (e.g., any two nodes can communicate with each other through multi-hop paths) [Di Francesco et al., 2011]. As a consequence, in most cases, the sensors are assumed to be static. In Mobile Wireless Sensor Networks (MWSN), a mobile element is introduced in order to improve the process of data collection and increase the lifetime of the network. Some of the works presented in the following sections use a direct communication among the mobile elements (mobile data collectors), and others works use an indirect communication among the fixed sensors and data's final destination (base station or sink).

3.1.1 Indirect communication protocols

Data MULE architecture [Shah et al., 2003] is a three-tier architecture for collecting sensor data; the approach takes advantage of the presence of mobile entities (called MULEs) present in the environment. When a MULE and sensor are in close range, MULEs pick up data from the sensor, buffer it and deliver the data to wired access points. The model assumes a two-dimensional random walk for mobility and incorporates key system variables such as a number of MULEs, sensors and access points. MULEs are assumed to be capable of *short-range* wireless communication, to have energy to travel through the sensing area and to exchange data from a nearby sensor or access point they encounter as a result of their motion. In this scheme, the sensors have to listen in order to identify a MULE's presence continuously. With multiple MULEs in the system, it is assumed that all the MULEs are performing independent random walks, with no communication among each other.

Zhao et al. [Zhao et al., 2004] presented a *Message Ferrying* approach which utilizes a set of special mobile nodes called *message ferries (MFs)* to provide communication service for nodes in the deployment area. Message ferries move around the deployment area and are responsible for carrying data among nodes. The main idea is to introduce non-randomness in the movement of the nodes. Message Ferrying protocol presents two varieties of the approach, depending on whether ferries or nodes initiate proactive movement. Ferries move around the deployed area according to known routes, collect messages from regular nodes and deliver messages to their destinations or other ferries. However, MFs know the location of sensor nodes. If the sensor nodes have knowledge ferry routes, they can adapt their trajectories to meet the ferries and transmit or receive messages.

In order to reduce the energy consumption in a cluster based sensor network, Zhang et al. [Zhang et al., 2009] proposed a dynamic data MULE path selection algorithm called Probabilistic Path Selection (PPS). In the cluster based network, the source node does not need to send data directly to the base station, the cluster heads (CHs) will store and relay the data, or CHs transfer data to data MULE when it arrives. In this work, a set of stationary nodes and one data-MULE are considered. All nodes were divided into clusters, and the distance between a source node and CH is one hop. The data-MULE travels along a fixed path and collects the data from the CHs and returns to the base station. During the runtime of PPS algorithm, the cluster heads are not changed. The PPS modifies the travel policy of the data MULE dynamically and ignores some CHs which have lower probability to have data. Nevertheless, nodes and data-MULEs need location information from each other, which is obtained from the GPS on the nodes or a location service in the network.

Yu et al. [Yu et al., 2008] proposed Grid-Based Mobile Element Scheduling (GBMES) schemes that schedule a mobile element (ME) to periodically gather data from a partially connected sensor network and any two of these fragments are disconnected from each other. First, the network is geographically partitioned into square grid cells. Then, in each grid cell a *MPR tree* rooted at the sensor node nearest to the geometric center of the grid cell is constructed. To collect data, a ME travels along a carefully designed route, and gathers sensed data from sensor nodes periodically. The sensor nodes are assumed to be static. The ME has sufficient energy, storage and processing capability. All sensors and the ME are aware of their own location through GPS signals or other localization approaches.

The Partitioning-Based Scheduling (PBS) algorithm is presented by Gu et al. [Gu et al., 2006]. They consider that the variety of data generation rates of sensors is an indicator to determine if some sensors need to be visited more frequently than others. In this work, all nodes are partitioned into several groups concerning their data generation rates and locations. Then, within a single group, the scheduling algorithm generates a node visiting priority for the mobile element (ME) to minimize the overhead for moving back and forth across faraway nodes. In this approach, ME needs *a-priori* knowledge about the location of the nodes. Finally, the scheduling solutions of the groups are concatenated forming the entire ME path so that all nodes can be visited at adequate frequencies to prevent any buffer overflow.

Somasundara et al. [Somasundara et al., 2004] also consider a network with static sensors in different areas operating at different sampling rates. This network is equipped with a mobile element (acting as a base station) that does the work of the data gathering. They consider that a node may need to be visited multiple times before all other nodes are visited depending on the strictness of its deadline (e.g., a frequency of sampling). This is called the Mobile Element Scheduling (MES) problem. As soon as a node is visited, its deadline (e.g., the time before which it should be revisited to avoid buffer overflow) is updated. Thus deadlines are “dynamically” updated as the mobile element performs the work of data gathering.

Van Le et al. [Van Le et al., 2014] proposed an architecture called Hierarchical Cooperative Data Gathering Architecture (HCDGA). HCDGA uses two types of mobile elements, MDCs and mobile relay (MR). The MR takes the collected data from the MDCs and delivers it to the sink. In this solution, first, the sensors are divided into one hop clusters. Cluster heads (CHs) are considered a meeting point for the MDCs. Each cluster has a MDC that collects their data periodically in a scheduled way. The MR regularly travels from the sink to visit some points of interest, called meeting points (MPs), to receive data from MDCs, and then returns to the position of the sink. HCDGA includes the ILP (Integer Linear Programming) to find the optimal trajectories of MDCs and the MR. However, both MDCs and MR do not share environmental information, and thus, dynamic conditions are not tackled for this work.

3.1.2 Direct communication protocols

Chakrabarti et al. [Chakrabarti et al., 2003], explore an alternative to saving power in sensor networks based on predictable mobility of the observer (or data sink). Predictable mobility is a good model for public transportation vehicles (buses, shuttles and trains), which can act as mobile observers. In this work, the sensor nodes are uniformly scattered over the area. At the beginning of the communication protocol, neither the observer knows nothing about the position of the individual sensors nor sensors know nothing about the path of the observer. However, due to repeated movements of the observer in the same path, the observer and sensors exchange information that helps them to acquire such knowledge about each other. In that moment, the observer has accurate knowledge about the positions of different sensor nodes. When the observer traverses the path, the data is *pulled* by the observer by waking up the nodes when it is close to them. This protocol needs knowledge about the sensor nodes location, and the data is collected indiscriminately. Besides, since the observer is a sink instead of a carrier of data, the communication, between the observer and the sensors, is performed in a direct fashion.

Faiçal et al. [Faiçal et al., 2014] presented an architecture to address the problem of self-adjustment of the UAV (Unmanned Aerial Vehicle) routes when spraying chemicals in a crop field. The algorithm readjusts the UAV path according to the data obtained from the wireless sensor network deployed in the crop field. Periodically, the UAV sends broadcasts messages to the sensors in the field to determine the amount of chemicals being perceived. If the sensor receives a message, it responds to the UAV with a message reporting the amount of measured chemicals and its position. On the basis of this information, the UAV can make a decision about whether to change its route or not.

Jea et al. [Jea et al., 2005] presented a solution for data collection using multiple mobile elements (data-MULEs). In particular, they present a load balancing algorithm which tries to balance the number of sensor nodes that each mobile element must visit. Their scenario considers static sensor nodes uniformly deployed in the area. With this approach each data-MULEs retrieved data approximately from the same number of nodes, considering shareable nodes among the MULEs. Initially, the MULEs make a round broadcasting of the beacons. After the sensors replay, each MULE has a list of nodes at a hop distance. However, a leader MULE (chosen previously) has the lists of all data MULEs. The leader MULE assigns the sensors' list that other MULEs must attend. With the assignment done, the data MULEs traverse their paths, polling for data.

3.1.3 Discussion

The works presented above take advantage of the fixed sensors' deployment in the sampling area. In most of these approaches, the knowledge of the location of the sensor is essential to data-collection. Therefore, a device or service to get the location of the nodes is included (e.g., GPS). Besides, many of the works take to MDCs as mechanical carriers of data, which pull the data from sensors and carry them to the sink (final destination of the collected data) regardless of whether it is useful or not. Some other works consider the generation rates of data to establish a scheduled visit to sensors. But if the sensor energy depletes then the covered area by that sensor is not covered anymore.

Even though there are solutions that use multiple MDCs, the MDCs are considered as mechanical carriers of data as well. The deployed sensors are grouped and an MDC is assigned to collect their data. In approaches like HCDGA [Van Le et al., 2014], the MDCs do not exchange their discoveries. Other approaches assumed a direct communication among the MDCs [Jea et al., 2005] to share their location but it is not capable of attacking dynamic conditions.

Finally, in protocols like those presented by Jea et al. [Jea et al., 2005] and Chakrabarti et al. [Chakrabarti et al., 2003] the MDC acts as a sink. Thus, keeping a direct communication between the sensor and the MDC is necessary. In this kind of scenarios, the sensors need to listen to MDCs continuously or learn their routes.

3.2 Data gathering from the environment

In this section, the related work in literature where mobile data collectors (MDCs) are used to take samples directly from the environment is reviewed. Thereby, these solutions do not depend on static sensors scattered in the sensing area. Hence, they do not have *a-priory* knowledge about the location of the data sources.

3.2.1 Direct communication protocols

Considering that regional surveillance refers to a continuous, repetitive and comprehensive search on a specified area in order to obtain essential information, Qu et al. [Qu et al., 2015] presented a solution of multi-UAV network for this issue. They take inspiration from the communication among ants through laying pheromones. The role of the pheromones in this paper is to guide the flight direction of the UAVs to obtain information quickly. For this, the pheromones increase and spread to adjacent regions. With the growth and spread of the pheromone, the UAVs fly to the pheromone saturated regions. However, if a pheromone is released or if a region is visited then this information is shared with the UAVs directly. It means that the UAVs are aware of the environmental changes.

Datataxis [Lee et al., 2009] is presented as a data harvesting algorithm in vehicular networks. Lee et al., designed agents, called MobEyes agents (e.g., police cars), capable to move around sampling area and harvest data from the regular vehicles when they are in direct communication range. The collected data have features about sensed data and context information such as timestamp and location. Regular cars collect data from other vehicles encountered opportunistically. MobEyes harvesting agents adapt their behavior by following a transition diagram that sometimes forces them to change their area of exploration. A regular node periodically warns newly generated data to its neighbors in order to increase the opportunities for agents to harvest the data. Depending on the mobility and the encounters of regular nodes, the packets are opportunistically diffused into the network of vehicles. The MobEyes agents may collect data from regular nodes by periodically querying the nearby nodes. The goal for the data harvesting is to collect all data generated in a specified area. The agent should harvest only those data packets that have not been collected already. The agents can infer that there may be other agents if the information density is lower than usual or significantly drops suddenly or a harvesting agent leaves a trail on the regular vehicles when it collects data. Each regular vehicle records this trail information which is returned to a newly encountered agent. The network is composed by various elements that permit to the MobEyes to get knowledge about the environment and accomplish the data harvesting task. This solution takes advantage of the elements (regular vehicles) and the deployed sensors in the area.

Stranders et al. [Stranders et al., 2013] present a near-optimal multi-agent algorithm for continuously patrolling such environment. The agents move on a graph, while taking measurements with the aim of maximizing the cumulative discounted *observation value* over time. The observation value is an abstract measure of reward, which encodes the properties of the agents' sensors, and the spatial and temporal properties of the measured phenomena. The optimal patrolling policy proposed is a set of observations that can be collected by the agents, subject to movement and observation constraints. This means that the agents share their information on a direct manner, thus, they used an enduring links communication.

Panait and Luke [Panait and Luke, 2004] proposed a pheromone-based algorithm for artificial agent foraging. The model allows the use of multiple pheromones, and an agent's choice of pheromones to update or to use in decision-making is based on the agent's current internal state. The algorithm uses two pheromones: p_{food} increases with proximity to the food sources, and p_{nest} increases the proximity to the nest. When the ant reaches its goal, it receives a positive reward. In this approach, the ants (agents) leave the nest while depositing the to-nest pheromone (released pheromones that indicate the path back to the nest). If the ants discover a food source, they begin the return to the nest along to the to-nest pheromone while depositing the to-food pheromone. In this way, the trail is established. Subsequently, all ants might be now engaged and finally, the ants perform trail optimization.

3.2.2 Indirect communication protocols

Data Foraging-Oriented Reconnaissance algorithm [Castañeda Cisneros et al., 2017] presented by Castañeda et al. is inspired by the stigmergy principle. They proposed an approach using a single MDC. Through release of pheromones in the environment, the MDC creates several paths and explores an operational environment with limited movement capabilities. They do not consider energy capabilities to traverse the whole sensing area in one trip. Due to this restriction, MDC (e.g., the aerial vehicle) needs to be recharged as many times as needed at a base station. Through various trips, the MDC can identify a region which has something of interest to the application. At the beginning, there is no information about the sampling area. Once the MDC has visited a region in the hextille grid representation of the sampling area, the MDC stamps the region. The number of stamps in a region indicates the number of visiting times. The objective of the reconnaissance is to expand the knowledge of the sampling area while visiting nodes. In order to explore new nodes getting to farthest and less stamped nodes is preferred.

3.2.3 Discussion

The related works presented in this section consider dynamical features in the environment. These solutions do not need *a-priori* knowledge about the location of the data sources. However, since the main objectives of these solutions are not focused in the design of a communication protocol, the solutions assume a direct communication among the agents or another strategy to share the global view of the environment. Despite considering an indirect communication protocol, the approach proposed by Castanẽda et al. [Castañeda Cisneros et al., 2017] uses a single MDC to perform the reconnaissance of an area.

3.3 Summary of related work

Many works near to data-foraging are proposed into Sparse Wireless Sensor Networks (SMWSN). Most of them take advantage of the knowledge of static sensors location, where the MDCs collect the data of all sensors in an indiscriminate fashion. In these kind of approaches it is difficult to adapt to dynamic conditions of the environment. Even with multiple MDC gathering data in the sampling area, a collaborative performance is not achieved because the MDCs do not exchange information about useful data sources. Other proposed solutions assume direct communication among the participant entities, without restrictions of their capabilities (e.g., range communication, storage, energy, etc.). Thus, an enduring transmission link is required.

Table 3.1 shows the principal features of the related work in which a single MDC is used. The majority of them need the *a-priory* knowledge of the data sources location. These works do not consider dynamic conditions in the environment. On the other hand, Table 3.2 shows the principal features of the proposed solutions in the literature that use multiple MDCs. In these works, it is assumed that the MDCs have endurable transmission links to share information among them. However, in a SMWSN it is difficult to keep these links of communication due to the dispersal of the MDCs.

Title	Dynamic environment	A-priory knowledge	Resources constraints	Mechanical carrier	Data selection	Communication paradigm
Data-MULE.Shah et al. (2003)	No	No	No	Yes	No	Indirect
Message ferries. Zhao et al. (2004)	No	Yes	No	Yes	No	Indirect
GBMES. Yu et al. (2008)	No	Yes	No	Yes	No	Indirect
PPS. Zhang et al. (2009)	No	Yes	No	Yes	No	Indirect
PBS. Gu et al. (2006)	No	Yes	No	Yes	No	Indirect
Predictable observer. Chakrabarti et al. (2003)	No	Yes	No	Yes	No	Direct
MES. Somasundara et al. (2004)	No	Yes	No	No	Yes	Direct
UAV for spraying pesticides. FaiÃ§al et al. (2014)	Yes	Yes	No	No	Yes	Direct
Data Foraging-Oriented Reconnaissance. Castañeda et al. (2017)	Yes	No	Yes	No	Yes	Indirect

Table 3.1: Comparative table of related work with a single MDC

Title	Dynamic environment	A-priory knowledge	Resources constraints	Mechanical carrier	Data selection	Communication paradigm	Data context
Data-MULE.Shah et al. (2003)	No	No	No	Yes	No	Indirect	Spatial
Multiple controlled data-MULE. Jea et al. (2005)	No	Yes	No	Yes	No	Direct	Spatial
HCDGA. Van Le et al. (2014)	No	Yes	No	Yes	No	Direct	Spatial
UAV for surveillance. Qu et al. (2015)	Yes	No	No	No	Yes	Direct	Spatial
Datataxis.Lee et al. (2009)	Yes	No	No	No	Yes	Direct	Spatio-temporal
Near-optimal continuous patrolling. Stranders et al. (2013)	Yes	No	No	No	Yes	Direct	Spatio-temporal
Pheromone-based algorithm for collaborative foraging. Panait et al. (2004)	Yes	No	No	No	Yes	Direct	Spatial

Table 3.2: Comparative table of related work with multiple MDCs

Chapter 4

Indirect spatio-temporal communication protocol

According to our methodology, an indirect spatio-temporal communication protocol to exchange information among devices of Sparse Wireless Sensor Network is presented. The protocol is inspired by the biological principle of stigmergy through secretion of pheromones. Artificial pheromones act as an indirect communication media to share information viewed by the mobile elements; thus the devices do not depend on enduring transmission links. The artificial pheromone is modeled as an abstract data type that allows a spatio-temporal communication without global references or synchronized physical clocks.

In this Chapter, first Sparse Mobile Wireless Sensor Network (SMWSN) is modeled as a distributed system and their elements are described. Later, the model of the sampling area is detailed. Finally, the artificial pheromones, as indirect communication method, and their operations are defined.

4.1 System model

The Sparse Mobile Wireless Sensor Network (SMWSN) is modeled as a distributed system (DS) specified mainly by processes and events.

- **Processes.** In a generic DS the processes are programs or instances of programs running simultaneously with other programs. Therefore, in an

SMWSN the entities associated with the system, the Base Station (BS) and the Mobile Data Collectors (MDCs), are represented as processes. Each process communicates with another process by message passing, sending one message at a time. Each process belongs to the set $P = \{p_1, p_2, ..., p_n\}$

  - **Base station**. The Base Station (BS) is a fixed node placed in the center of the sensing area. It is assumed that BS has enough storage and computational resources. Let bs be the base station of the system, $bs \in P$.
  - **Mobile data foragers (MDF)**. There are those sensors with mobility capabilities. Each MDF belongs to the set MDF=$\{mdf_1, mdf_2, ..., mdf_v\}$, with $MDF \subset P$. The MDFs do not have direct communication among themselves, they exchange messages through the base station. Each $mdf \in MDF$ has limited energy and computational resources.

- **Messages**. Messages are abstractions to represent frames or packets in a computer network, which contain arbitrarily complex data structures. For our approach, each exchanged message in the system belongs to the set M. A message $m \in M$ is a tuple $(smdf, payload)$, where $smdf$ is the identifier of the process that originally generates the message and $payload$ is a composite data structure.

- **Events**. An event represents an instant execution performed by a process. We consider two kinds of events:

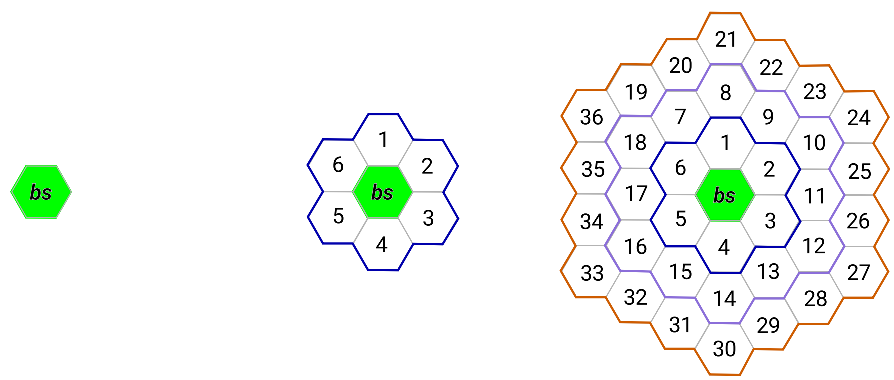
  - *send* refers to the emission of a message executed by a process.
  - *receive* refers to the reception of a message in a process.

- **Operational area (OA)**, also called sensing area, refers to a bound surface where MDFs collect environmental data.

4.2 Modeling the operational area

The operational area (OA) is geometrically modeled as a discrete and finite two-dimensional tessellation formed by regular hexagons, called hextille. Each hexagon of the hextille represents a limited region r of the OA. Thus, the OA is the set $OA = \{r_0, r_1, r_2, ..., r_u\}$. For this work, they are considered symmetrical hextilles with $3h^2 - 3h + 1$ hexagons, where $h = \{1, 2, 3, ...\}$. It is said that h denotes the degree of the hextilles (see Figure 4.1).

For practicality, the central hexagon of the hextille is reserved to host the base station bs. The hextille can be seen as a central hexagon surrounding by $h - 1$ rings of hexagons. Each ring of hexagons represents a level l of OA, composed of $6l$ hexagons.



(a) OA_1 **(b)** OA_2 **(c)** OA_4

Figure 4.1: Operational area modeled as hextille of degree a) h =1, b) h =2 y c) h =4.

The regions are numbered in a clockwise direction beginning with the central hexagon, which has assigned 0, and continued with the region above of this, as depicted in Figure 4.2.

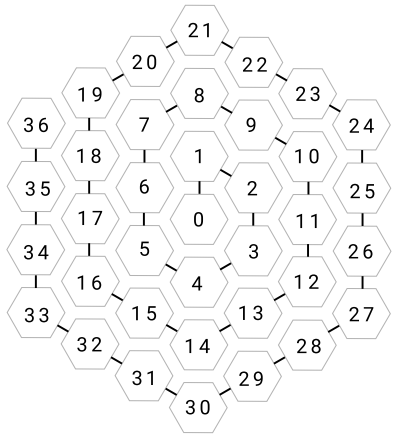


Figure 4.2: Numeration of the regions of OA_4

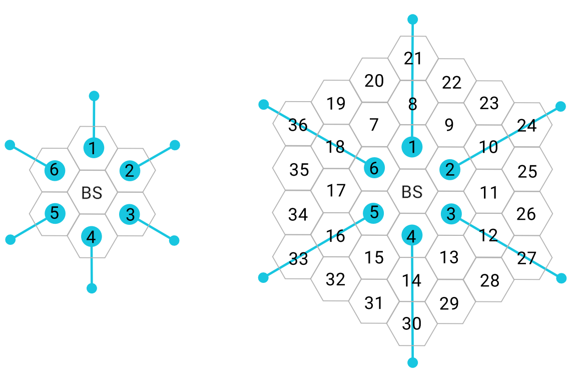
4.2.1 Sub-operational areas

To reduce the time to traverse the operational area, OA is divided into sub-operational areas $S_{oa} \subset OA$ with the same number of regions:

$$|S_{oa}| = \frac{h^2 - h}{2} \tag{4.1}$$

The regions $r \in S_{oa}$ are chosen according to the following rules:

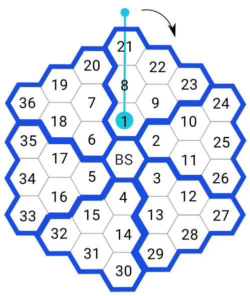
- The adjoining regions to r_0 are used as the axis of reference to dividing OA into equivalent and symmetric six parts. Figure 4.3 depicts in blue lines these axes of reference.



(a) OA_2 **(b)** OA_4

Figure 4.3: Axis of reference to divide the hextille into S_{oa}

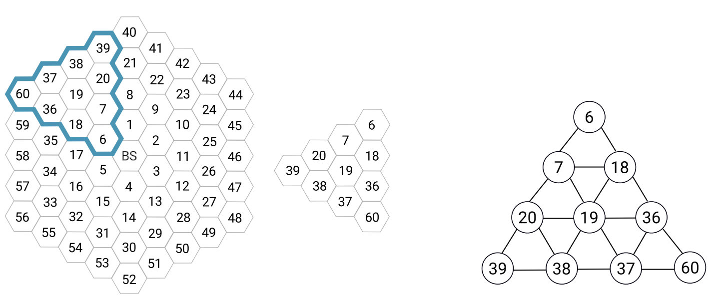
- From the axis of reference, the MDF choose if the S_{oa} to visit is built in dextrorotation (Figure 4.4a) or levorotation(Figure 4.4b) way. Considering this, there are $\{S_{oa_1}, S_{oa_2}, \cdots, S_{oa_{12}}\}$ to assign.



(a) (b)

Figure 4.4: OA divided into a) **dextrorotation** and b) **levorotation** S_{oa} as from the same axis of references

- A subarea S_{oa} can be modeled as an undirected graph where each node has a maximum of six neighbors. Nodes are related by edges if the regions share a vertex. Figure 4.5 depicts an S_{oa} of an OA_5 with ten regions. Each $S_{oa} \in OA_5$ will have the same number of regions and a graph with the same form that the shown in Figure 4.5b. In general, per each level of the graph, the number of nodes increases in one. It means, the number of regions per levels of S_{oa} is $\{1, 2, \cdots, h-1\}$.



(a) (b)

Figure 4.5: From a) subarea S_{oa} in dextrorotation to b) graph.

4.3 Artificial pheromone

In the next Section, artificial pheromones are defined as a mechanism to allow indirect communication among MDFs. The operations of *create* and *decay* are modeled through causal and fuzzy causal dependencies. Also, accumulated pheromones in a region generate an intensity, which is used to determine the "freshness" of the observations taken by the MDFs.

An artificial pheromone f is modeled as an abstract datatype, which is used as a mechanism to perform indirect communication among MDFs through the BS. Taking inspiration on the biological principle of stigmergy, a pheromone is used to exchange a message among involved entities.

Definition 4.1. *Formally, an artificial pheromone is defined as the tuple:*

$$f_{(u,v)} = \{r_u, mdf_v, ts, V_{trail_b}, V_{trail_f}\}$$

where

- r_u is the identifier of a region in which the pheromone was secreted.
- mdf_v is the identifier of the MDF that deposited the pheromone.
- ts is a timestamp that denotes the number of regions visited before the MDF reaches the region r_u.
- V_{trail_b} is the set of regions from the bs to r_u. These regions indicate the backward path to bs.
- V_{trail_f} is the set of regions visited after secreting the pheromone in r_u. These regions indicate a straight path to the food.

4.3.1 Secretion of pheromones

When an MDF finds a possible data source, a pheromone f is released. Thereby, the value of ts is the number of regions visited before reaching a region r_u. The maximum number of regions visited before the MDF returns to bs depends on $|S_{oa}|$. The transition movement of a region r_u to any adjacent region r'_u by an MDF is called a **step**. The total of steps to travel S_{os} is calculated by Equation 4.2:

$$steps = \frac{h^2 - h + 4}{2} = |S_{oa}| + 2 \tag{4.2}$$

Once that the MDF arrives at bs, the information collected by an MDF must be shared with the other. As distributed entities, each MDF has its own timeline. Therefore, when an MDF arrives at the base station, its local timeline must be aligned to the global timeline of the BS.

4.3.1.1 Global temporal alignment

The base station has its own timeline, called global timeline, which is coordinated with a physical clock. The global timeline establishes time windows of a step size to define a clock tick. So when an mdf arrives at bs, its recorded pheromones must be aligned to this global timeline. For that, it is necessary to determine the number of steps from the pheromone released to the end of a trip, which is calculated by Equation 4.3:

$$CD = steps - ts \tag{4.3}$$

CD represents the causal history of the pheromone, it means, the number of regions visited before to return to bs. With CD, the timestamp according to global time is estimated by Equation 4.4:

$$tsg_{f_{(u,v)}} = at_{mdf} - CD \tag{4.4}$$

where at_{mdf} is the arrival step according to bs and $tsg_{f_{(i,j)}}$ is the global step of pheromone secretion. Figure 4.6 depicts an example of this global alignment with three pheromones. The vertical lines in color blue, orange and red; represent the arrival step to bs of the MDFs mdf_1, mdf_2, mdf_3 and the circles of their pheromones. Each MDF has a total of steps per trip $= 6$. The orange circle represents a pheromone secreted in r_{18} by mdf_2, which was release in step 2 of a trip. However, this MDF arrives at bs in the step $= 7$ of the global timeline (at_{mdf}). With this information $CD = 6 - 2$ and $tsg_{f_{(18,2)}} = 7 - 4 = 3$, the mapping is carried out.

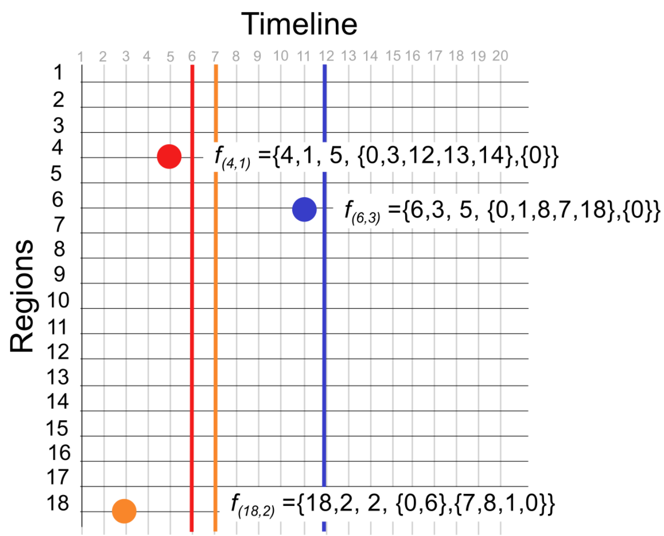


Figure 4.6: Global temporal alignment of pheromones

Once those pheromones are aligned, it is possible to observe that the pheromones were secreted in different steps. Therefore, some of these pheromones are older than the others. In the following Section, the calculation of this "decay" is detailed.

4.3.2 Oldness of pheromones

Each secreted pheromone represents an observation taken in a region with possible profitable data-source. Naturally, the freshness of the observations took by the

MDFs decreases while the time elapses, it means that the pheromones age. It is necessary to consider a reference to determine how ancient a pheromone is. In this approach, the reference is the arrival of MDF that traverse the last S_{oa} assigned.

4.3.2.1 Final arrival time of the MDFs

Considering that each recorded clock tick is the transition from a region to another, the number of steps required by an MDF to traverse a S_{oa} can be estimated (See Equation 4.2). Depending on the number of MDFs into the system and taking into account that the MDFs takeoff of bs in a sequential manner, the required steps to traverse all S_{oa} can be estimated as indicated in Equations 4.5a, 4.5b and 4.5c:

$$at_{final} = \begin{cases} steps\dfrac{12}{|MDF|} + |MDF| - 1 & \text{IF } 12 \ \% \ |MDF| = 0 \quad (4.5a) \\ steps\lceil\dfrac{12}{|MDF|}\rceil + (12\%|MDF|) - 1 & \text{IF } 12 \ \% \ |MDF| \neq 0 \quad (4.5b) \\ steps + 12 - 1 & \text{Otherwise} \quad (4.5c) \end{cases}$$

where at_{final} is the final arrival time that is used as a reference to calculate the oldness, and 12 is the number of $S_{oa} \in OA$. Three cases can be observed for at_{final}:

- Equation 4.5a: Each $mdf \in MDF$ performs the same number of trips.
- Equation 4.5b: Some MDFs performs more trips than others.
- Equation 4.5c: There are several MDFs and not all of them have departures.

Being at_{final} the reference, the oldness of a pheromone is calculated as follows:

$$old_f = at_{final} - tsg_{f_{(u,v)}} \tag{4.6}$$

4.3.3 Intensity

In the subsection 4.2.1, the operational area OA is divided into subareas S_{oa}. However, six of them are in dextrorotation and six in levorotation, so that some subareas are overlapped. In nature, the number of pheromones in the same place can intensify the trail of food. Hence, the intensity of artificial pheromones is determined by relating the oldness and the number of pheromones in the following way:

"how old and how dense a trail is implies how intense it is"

The intensity of the secreted pheromones in a region can be estimated using a Fuzzy Inference System (FIS) based on the fuzzy-causal relation (FCR) [Perez Cruz and Pomares Hernandez, 2014].

4.3.3.1 Fuzzification process

The intensity of the pheromones deposited in a region represents a degree of freshness of the observations, considering the information about the logical/temporal distance of the events' sources, similarly to the *degree closeness* of the FCR.

To achieve this, we define the following linguistic variables (see Definition 2.6):

- For the Accumulated Oldness (AcO), whose universe of discourse is the average of the oldness of the pheromones secreted in a region.
- For the Amount of pheromones (AP), whose universe of discourse is the amount of secreted pheromones in a region.
- For the Intensity per region (IPR), whose universe of discourse is the intensity of the trail of pheromones in a region.

The inputs and outputs are fuzzified using a triangular function fuzzifier (see Definition 2.3). Five triangular membership functions related to five linguistic terms are defined for the linguistic variables.

- **Accumulated Oldness**: VR related to *VeryRecent*, R related to *Recent*, MR related to *Medium Recent*, A related to *Ancient*, VA related to *VeryAncient*.

- **Amount of pheromones**: L related to *Little*, F related to *Few*, E related to *Enough*, M related to *Many*, S related to *Several*.

- **Intensity per region**. L related to *Lack*, W related to *Weak*, M related to *Medium*, S related to *Strong*, VS related to *VeryStrong*.

The different fuzzy sets are delimited as shown in Table 4.1. The values σ_0, σ_8, θ_0, θ_8, ψ_0 and ψ_8 must be chosen using previous knowledge about the environment's conditions. Thus, σ_0, σ_8 are determined by the minimum or maximum average of the oldness of the secreted pheromones in a region, which is calculated by the Equation 4.5a; $\theta_0 = 0$ and $\theta_8 = 4$ are the minimum and the maximum of pheromones in a region r_u due to the maximum number of pheromones secreted in the nearest regions to bs; while ψ_0 and ψ_8 take the scalar value between 0 and 100.

Table 4.1: Values of variables used in definition of membership functions.

Universe of discourse	Set	a	b	c
Accumulated Oldness	VR	$\sigma_0 - \sigma_2$	σ_0	σ_2
	R	σ_0	σ_2	σ_4
	MR	σ_2	σ_4	σ_6
	A	σ_4	σ_6	σ_8
	VA	σ_6	σ_8	$\sigma_8 + \sigma_2$

Continued on next page

Table 4.1 – *Continued from previous page*

Universe of discourse	Set	a	b	c
Amount of pheromones	L	$\theta_0 - \theta_2$	θ_0	θ_2
	F	θ_0	θ_2	θ_4
	E	θ_2	θ_4	θ_6
	M	θ_4	θ_6	θ_8
	S	θ_6	θ_8	$\theta_8 + \theta_2$
Intensity per region	L	$\psi_0 - \psi_2$	ψ_0	ψ_2
	W	ψ_0	ψ_2	ψ_4
	M	ψ_2	ψ_4	ψ_6
	S	ψ_4	ψ_6	ψ_8
	VS	ψ_6	ψ_8	$\psi_8 + \psi_2$

Figures 4.7, 4.8 and 4.9 depict the fuzzy set described in Table 4.1. These Figures are the graphical representation of the fuzzification process. In each Figure, the linguistic term for each linguistic variable is defined.

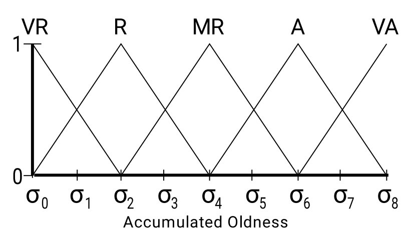


Figure 4.7: Input fuzzy sets of the oldness of the pheromones secreted in a region.

No.	Rule IF-THEN	No.	Rule IF-THEN
1	If AcO is VR and AP is L then IPR is M	14	If AcO is A and AP is E then IPR is M
2	If AcO is R and AP is L then IPR is M	15	If AcO is VA and AP is E then IPR is W
3	If AcO is MR and AP is L then IPR is W	16	If AcO is VR and AP is M then IPR is S
4	If AcO is A and AP is L then IPR is W	17	If AcO is R and AP is M then IPR is S
5	If AcO is VA and AP is L then IPR is L	18	If AcO is MR and AP is M then IPR is M
6	If AcO is VR and AP is F then IPR is S	19	If AcO is A and AP is M then IPR is M
7	If AcO is R and AP is F then IPR is M	20	If AcO is VA and AP is M then IPR is W
8	If AcO is MR and AP is F then IPR is M	21	If AcO is VR and AP is S then IPR is VS
9	If AcO is A and AP is F then IPR is W	22	If AcO is R and AP is S then IPR is S
10	If AcO is VA and AP is F then IPR is W	23	If AcO is MR and AP is S then IPR is S
11	If AcO is VR and AP is E then IPR is S	24	If AcO is A and AP is S then IPR is M
12	If AcO is R and AP is E then IPR is S	25	If AcO is VA and AP is S then IPR is M
13	If AcO is MR and AP is E then IPR is M		

Table 4.2: Inference rules

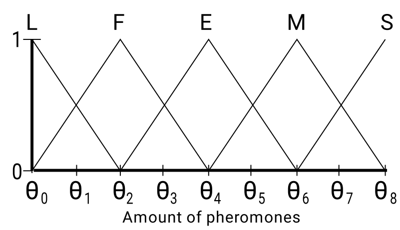


Figure 4.8: Input fuzzy sets of the amount of the pheromones secreted in a region

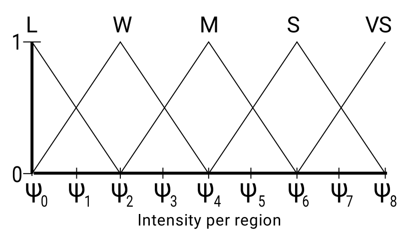


Figure 4.9: Output fuzzy sets of the intensity per region

4.3.3.2 Fuzzy inference

To determine the degree of the intensity in a region, the fuzzified inputs and outputs are related through a Mamdani-type FIS (Section 2.5). This Mamdani-type FIS consists of 25 IF-THEN rules as shown in Table 4.2.

Since the outputs of inference system are fuzzy output variables, it is necessary to convert the fuzzy output variables into crisp values through a defuzzification process. The defuzzification process chosen is the Weighted Average method (see Section 2.4) since it has low computational cost compared with others methods. Since symmetrical membership functions were defined in the fuzzification process, the Weighted Average method results on a good choice. Therefore, the overall input-output surface corresponding to the above membership functions, values of variables, and rules are depicted in Figure 4.10. This surface shows that if in a region the oldness increases and the number of pheromones is little, the intensity for that region is scarce. It means, at the higher freshness and more pheromones, the intensity is stronger.

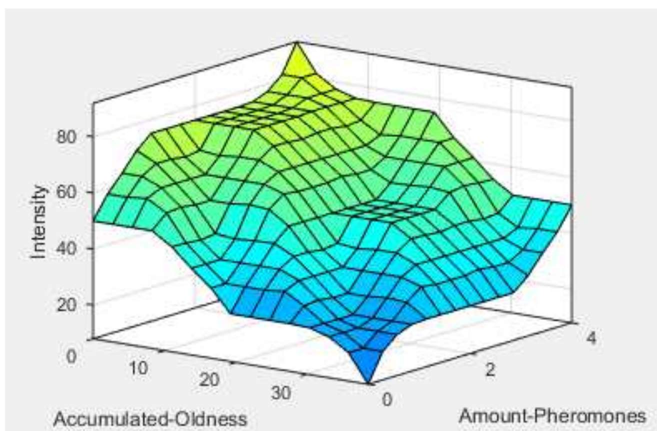


Figure 4.10: Output fuzzy sets of the intensity per region

After the defuzzification process, the IPR represents the intensity of the secreted pheromones in a region. Now, the MDFs know about of the environment through the intensity of the regions. And this information can be shared to all MDFs in the system.

Chapter 5

Collaborative data-foraging mechanism

This Chapter presents a mechanism to search and retrieve profitable data sources from dynamic environments with a collaborative approach. This mechanism implements the indirect spatio-temporal communication protocol described in Chapter 4. The aim is tackling the lack of enduring links among the devices in a Sparse Wireless Sensor Network (SMWSN). Next, the algorithms of reconnaissance, identification of regions of interest and data-harvesting stages are described.

5.1 Definition of the distributed data-foraging problem

Collaborative data-foraging consists of a set of activities, executed by multiple mobile nodes in a distributed manner. These tasks are performed within an **Operational Area** (OA) (see Section 4.2) in which not all regions provide relevant information. Below the distributed data-foraging problem is defined:

> ***Distributed data foraging (DDF) problem for dynamic environment.*** *Given an operational area composed by* $\{r_1, r_2, \ldots\}$*, data foraging is collaboratively executed by a set of mobile nodes* $\{mdf_1, mdf_2, \ldots\}$ *through a cyclical process composed of three main tasks: reconnaissance, identification*

of regions of interest and data-harvesting. These tasks must be completed within a certain time, accomplishing the temporal constraints of data.

Into SMWSN the activities for DDF are described as follows:

- **Reconnaissance.** The objective of the reconnaissance is visiting, at least once, each region of OA in search of possible profitable data sources. Due to their limited energy and computational resources, the mobile nodes visit a certain number of regions in each trip before returning to the base station. Therefore, more than one trip is necessary to visit the whole OA. Each region r becomes in an inspection region r_{insp} if relevant information is found in it. The worth of the found information depends on the application.

- **Identification of regions of interest.** Once the whole OA has been visited, it is necessary to perform a weighted oversampling of the inspection regions $R_{insp} = \{r_{insp_1}, r_{insp_2}, \dots\} \subseteq \{r_1, r_2, \dots\}$ to identify such regions which have the most profitable data. The mobile nodes must make more than one trip to obtain oversampling in such regions. Those regions with the most profitable data compose the set of regions of interest $ROI = \{roi_1, roi_2, \dots\}$.

- **Data Harvesting.** The objective of data harvesting is the retrieving of data of the regions of interest $ROI \subseteq R_{insp}$.

5.1.1 Proposed solution to the DDF problem

The DDF problem requires a collaboration among the mobile nodes to perform reconnaissance, identification of regions of interest and data-harvesting. The tasks involve with the DDF are comparable to the behavior of the social insect societies in nature for searching of nutrient sources. Finding food sources, communicating their location to the other members of the society and collecting the food are activities for each member of a group foraging.

In this work, to accomplish the objective of each activity of DDF, a mechanism of collaborative data-foraging inspired in the social insect societies is proposed. The mobile entities that execute those activities are called mobile data foragers (MDFs). The mechanism is composed of different algorithms that integrate the indirect communication protocol to achieve a collaborative performance of the MDFs.

5.2 Reconnaissance

This section describes a mechanism related to the task of reconnaissance. At the beginning of a data-foraging cycle, there is no information about the location of the data sources into an operational area (OA). Therefore, the Mobile Data Foragers (MDFs) visit each region of OA to find it. A heuristic is proposed to control the mobility of the MDFs during their trips considering energy restrictions. During a trip, the *mdf* can release artificial pheromones if possible profitable data sources are found. Through pheromones, the location of those data sources are shared among the entities.

5.2.1 Mobility of the MDF

Mobile Data Foragers traverse the operational area to sample the environment in search of useful data sources. Unlike random mobility, a controlled movement becomes an additional factor which can be exploited in MWSN. In the case of collaborative data-foraging, an MDF does not need to know the whole operational area. Thus, a travel heuristic is proposed in order to define the MDFs trajectory to traverse a sub-operational area (see Section 4.2.1).

There are some considerations to take before to the MDFs begin the first task of the collaborative data-foraging:

- After the OA with degree h has been selected, it is divided into a set of sub operational areas $SOA = \{S_{oa_1}, S_{oa_2}, \ldots, S_{oa_{12}}\}$ with the same number of regions (Equation 4.1). Where the detection range of sensors of an MDF represents a region r of the OA.

- To avoid collisions, the MDFs departures from the base station bs are made in a consecutive fashion.

- Randomly, an $mdf \in MDF$ chooses a sub-operational area S_{oa} constructed either in a levorotatory or dextrorotatory manner (see Subsection 4.2.1). Each S_{oa} is chosen once.

- The trips of reconnaissance have a finite time of duration.

- The mdf only moves to adjacent regions r'_i to it placed (r_u). It means, there is no teleportation [Betke et al., 1995].

The heuristic proposed in this work gives priority to the farthest and no visited regions, as described as follows:

1. Given a region r_u with a set of neighbors N_{r_u}, the region to visit is $r'_u \in N_{r_u} | d(r'_u, bs) > d(N_{r_u} - r'_u, bs)$.

2. r'_u has fewer neighbors visited than others adjacent regions that belong to N_{r_u}.

3. If there are two or more adjacent regions that satisfy 1) and 2) choose the next one randomly.

The selection of the r'_i to visit is detailed in Algorithm 1. Among lines 1 - 15, the procedure that returns the r'_u to visit is described. First, given a region r_u the farthest adjacent region r'_u is chosen (lines 16 - 35). If there are two or more farthest regions unvisited then the adjacent region with less visited regions in it neighborhood is chosen, as is detailed among the lines 36 - 51. Finally, after $\forall r_u \in S_{oa}$ is visited, the MDF returns to bs.

Algorithm 1 Give the next region to visit

Input variables:

The adjacent regions of a region r_u: List<Region>*Neighbors*

Output variables:

The next region to visit: Region r'_u

Local variables:

The set of no visited neighbors of r_u: List<Region>*noVisited*

The set of farthest regions to the base station: List<Region>*farthest*

```
1: procedure ChooseNeighbor(Neighbors)
2:     List<Region>noVisited ← GetNoVisitedNeighbors(Neighbors)
3:     List<Region>farthest ← GetFarthest(noVisited)
4:     if farthest.length > 1 then
5:         List<Region>lessNeighbors ← GetLessNeighbors(farthest)
6:         if lessNeighbors.length > 1 then
7:             r'_u ← random(lessNeighbors)
8:         else
9:             r'_u ← lessNeighbors[0]
10:        end if
11:    else
12:        r'_u ← farthest[0]
13:    end if
14:    return r'_u
15: end procedure
```

Return the neighbors that have not been visited of a determinate region

```
Input variables:
- The adjacent regions of a region r_u: List<Region>Neighbors
Output variables:
- The set of no visited adjacent regions of r_u: List<Region>noVisited
Local variables:
- The current region being checked: Region r'_u
16: function GetNoVisitedNeighbors(Neighbors)
17:     List<Region>noVisited ← ⊘
18:     for all r'_u ∈ Neighbors do
19:         if r'_u.visited = false then
20:             noVisited ← noVisited ∪ r'_u
21:         end if
22:     end for
23:     return noVisited
24: end function
```

Algorithm 1 **(Continue)** The next region to visit is chosen

Return the farthest neighbors to base station of a determinate region

```
Input variables:
- The no visited adjacent regions of a region r_u:  List<Region>noVisited
Output variables:
- The set of farthest regions:  List<Region>farthest
Local variables:
- The distance between the base station and the farthest neighbor of r_u:  int maxDistance
-The current region being checked:  Region r
function GetFarthest(noVisited)
    noVisited ← ∅
    maxDistance ← GetMaxDist(noVisited)
    farthest ← ∅
    for all r ∈ noVisited do
        if r.distance = maxDistance then
            farthest ← farthest ∪ r
        end if
    end for
    return farthest
end function
```

Return the neighbor that have not been visited with lesser visited neighbors

```
Input variables:
- A set of adjacent regions of a region r_u:  List<Region>Neighbors
Output variables:
- The adjacent region of a region r_u with lesser no visited neighbors:  Region r'_u
Local variables:
- The current region being checked:  Region r'_u
- The set of no visited adjacent regions of a r'_u:  List<Region>noVisitedN
- Minimum number no visited regions of a neighbor r'_u:  int number
- Set of neighbors with lesser no visited neighbors:  List<Region>nvNeighbors
function GetLessNeighbors(Neighbors)
    List<Region>noVisitedN ← ∅
    int number ← ∞
    for all r'_u ∈ Neighbors do
        List<Region>nvNeighbors ← GetnoVisitedNeighbors(GetNeighbors(r'_u))
        if nvNeighbors.length < number then
            List<Region>noVisitedN ← ∅
            number ← nvNeighbors.length
            noVisitedN ← noVisitedN ∪ r'_u
```

Algorithm 1 **(Continue)** The next region to visit is chosen

```
        else
            if nvNeighbors.length = number then
                noVisitedN ← noVisitedN ∪ r'_u
            end if
        end if
    end for
    if noVisitedN.size > 1 then
        r'_u ← Random(noVisitedN)
    end if
    return r'
end function
```

5.2.2 Pheromone secretion

During the reconnaissance trip, if an *mdf* find in a region r_u with valuable information to the application then that region is stamped with an artificial pheromone $f_{(i,j)}$. The pheromone is defined as the tuple $f_{(u,v)} = \{r_u, mdf_v, ts, V_{trail_b}, V_{trail_f}\}$ (see Subsection 4.3). Each $mdf \in MDF$ has its own logical timeline (local timeline) where each clock tick is represented by the pass of a r_u to another, in other words a clock tick is a *step* (see Subsection 4.3.1), and the timeline is not shared with the other MDFs.

The maximum steps of an MDF to traverse a S_{oa} is equal to the number of regions in the $S_{oa} + 2$ (as it is defined in Subsection 4.3.1). When a pheromone is secreted, ts corresponds to the current tick in the local timeline of the MDF. In addition, two trails before (V_{trail_b}) and after (V_{trail_f}) to release the pheromone are recorded. Algorithm 2 details the secretion of pheromones during the reconnaissance trip. In line 17 of Algorithm 2 a pheromone is secreted and from lines 10 to 14, in the same Algorithm, the trail V_{trail_f} is saved. From lines 28 to 36 each element of a pheromone is described.

Algorithm 2 Reconnaissance of a subarea

Input variables:

The set of regions of a designated subarea to visit: List<Region>*map*

Identifier of a MDF: int *id*

Output variables:

The set of pheromones secreted by the MDF: List<Pheromone>*secreatedPh*

Local variables:

The total number of steps to traverse an subarea: int *steps*

The local timeline: int *ts*

The current region being checked: Region *r*

The set of visited region before to reach a region with food: List<Region> *backTrail*

The set of visited region after to reach a region with food: List<Region>*frontTrail*

The set of adjacent regions of *r*: List<Region>*Neighbors*

```
procedure ReconMobility(map, id)
    int steps ← map.length + 2
    int ts ← 0                                        ▷ Local timeline
    Region r ← 0                                      ▷ All MDF take off Base Station
    List<Region>backTrail ← ∅
    List<Pheromone>secretedPh ← ∅
    while steps ≥ 0 do
        backTrail ← backTrail ∪ r                     ▷ Back trail before to reach a region with food
        if secretedPh ≠ ∅ then
            for all f ∈ secretedPh do
                f.frontTrail ← f.frontTrail ∪ r       ▷ Front trail after to secrete a pheromone
            end for
        end if
        r.visited ← true
        if r.hasFood then
            Pheromone f ← CreatePheromone(r, id, ts)  ▷ Create an artificial pheromone
            f.backTrail ← backTrail
            secretedPh ← secretedPh ∪ f
        end if
        List<Region>Neighbors ← GetNeighbors(r)       ▷ Give the neighbors of the current region
        r ← ChooseNeighbor(Neighbors)                 ▷ Give the next region to visit
        steps ← steps − 1
        ts ← ts + 1                                   ▷ Increase the local timeline
    end while
    return secretedPh
end procedure
```

Create a pheromone

```
function CreatePheromone(idRegion, idMDF, timeStamp)
    pheromone f
    f.id_r ← idRegion
    f.id_mdf ← idMDF
```

Algorithm 2 **(Continue)** Reconnaissance of a subarea

```
        f.ts ← timeStamp
        f.backTrail ← ∅
        f.frontTrail ← ∅
        return f
    end function
```

The first part of the collaborative data-foraging mechanism has been presented. The controlled movement to the MDF to traverse a sub-operational area S_{oa} improve the network performance. Besides, if a region with worthy information is found, an artificial pheromone $f_{(i,j)}$ is released. However, the released pheromones by one MDF are not sufficient to determine if a region provides a valuable data source. Thus, we take advantage of the overlap of $\{S_{oa_1}, S_{oa_2}, \ldots, S_{oa_{12}}\}$ and the view of different MDFs, due to that the collected samples taken in the regions where pheromones were released generate redundancy. In this way, it is possible to determine if a region contains a valuable data source.

5.3 Identification of regions of interest

After finishing the reconnaissance of all sub-operational areas SOA, a set of pheromones $Phero$ is delivered by the MDFs. In other words, once an *mdf* does its trips and returns to bs, it deposits and submits its observations. Later the *mdf* continues with another trip if it is required. At bs, when an observation in a region r_u has some value of interest, r_u becomes in a **region of inspection** r_{insp}. To record a common region as a region of inspection, a pheromone should be secreted. Each region of inspection belongs to the set $R_{insp} \subseteq R$.

5.3.1 Recording of pheromones

The regions of inspection r_{insp} were not visited at the same time, thus first the alignment with the global timeline of the bs (see Section 4.3.1.1 of Chapter 4) must be performed. This procedure is described in Algorithm 3. From the local view of MDF, the logical distance between the secretion of the pheromone and the end of the trip is calculated (line 3). The MDF arrival time to bs and the logical distance allows estimating the global time in which the pheromone was released (line 4). With this, the alignment is achieved.

Algorithm 3 Base Station records the pheromones segregated by a MDF

Input variables:
The set of released pheromones: List<Pheromone>$Phero_{mdf}$
Step in the global timeline in which the MDF landed : int $t_{landing}$
Output variables:
Local variables:
The current pheromone being checked: Pheromone f
The total number of steps to traverse an subarea: int *steps*
Step in which a pheromone was released: int $f.ts$
Distance between pheromone's release and landing step: int $lDistance$
Step in the global timeline in which the pheromone is recorded: int *globalTS*
Set of recorded pheromones by base station: List<Pheromone>$Phero_{bs}$

1: **procedure** RecordPheromones($Phero_{mdf}$, $t_{landing}$)
2:     **for all** $f \in Phero_{mdf}$ **do**
3:         int $lDistance \leftarrow steps - f.ts$ ▷ Logical distance
4:         int $globalTS \leftarrow t_{landing} - lDistance$
5:         $f.ts \leftarrow globalTS$ ▷ Update the timestamp of the pheromone
6:         $Phero_{bs} \leftarrow Phero_{bs} \cup f$ ▷ Save the updated pheromone in the base station
7:     **end for**
8: **end procedure**

5.3.2 Identification of ROIs

The result of the reconnaissance is a set of regions of inspection R_{insp}. Despite a region of inspection contains profitable information to the application, it is not identified as interesting yet. A region of inspection r_{inps} becomes a region of interest roi if the worth of the region is reinforced through secretion of pheromones.

As mentioned earlier, due to the SOA contains dextrorotation and levorotation subareas S_{oa} each r_u has overlap. This allows having redundancy of sampling information from the reconnaissance mechanism. The reconnaissance is performed by multiple MDFs, this implies that a region r_{inps} can have pheromones secreted at different steps. The intensity per each region with pheromones should be estimated in order to discard the regions $r_{insp} \notin ROI$.

To estimate the intensity of the pheromones secreted in a region, the oldness of each pheromone must be calculated, according to Equation 4.6 of Section 4.3.2. The oldness is calculated in line 3 of Algorithm 4. To calculate it, the step of bs at the moment to finish the reconnaissance is considered and timestamp pheromone in a global time.

Algorithm 4 Calculate the oldness of each pheromone

Input variables:

Global step in which the last MDF landed: int $globalTime$

Local variables:

The current pheromone being checked: Pheromone f

Deterioration of a pheromone: int old

Set of old values of all pheromones secreted in the same region: List$\langle old \rangle Oldness$

Set of regions of an Operational Area: List$<\text{Region}>R$

1: **procedure** CalculateOldness($globalTime$)

2:     **for all** $f \in Phero_{bs}$ **do**

3:         $old \leftarrow globalTime - f.ts$ ▷ Calculate the deterioration of a pheromone

4:         $R[f.id_r].Oldness \leftarrow R[f.id_r].Oldness \cup old$ ▷ Save the deterioration of all pheromones

5:     **end for**

6: **end procedure**

Once oldness for each pheromone is calculated, the intensity is estimated using the Fuzzy Inference System (FIS) mentioned in Section 4.3.3. The first part of the FIS is the fuzzification of the inputs. In this case, the input variables are the oldness average of the secreted pheromones (AO) and amount of them in a region (AP). The output variable is the estimated intensity per region (IPR). Algorithm 5 performs the operations to estimate the IPR. For each linguistic variable, five triangular functions are defined and the Mamdani rules defined in Table 4.1 are applied.

Among lines 6 and 9, the process to convert a crisp input into a fuzzy value is carried out. While from line 11 to 35 the 25 Mamdani rules described in Subsection 4.3.3.2 are presented. Finally, among line 36 and 40 the defuzzification process to return the fuzzy value in a crisp output is performed.

Algorithm 5 Estimate an intensity per region with pheromones

Input variables:

Average *old* of a region r: int $oldness_r$

Amount of pheromones secreted in a region r: int $amount_r$

Output variables:

The estimated intensity per region: double IPR

Local variables:

Fuzzy average oldness value in a region r: int OA

Fuzzy amount of pheromones value in a region r: int AP

---

procedure CALCULATEIPR($oldness_r$, $amount_r$)

    #define: $a_k^{AO}, b_k^{AO}, c_k^{AO}\ \forall k : 1 \ldots 5$ ▷ Five triangular functions

    #define: $a_k^{AP}, b_k^{AP}, c_k^{AP}\ \forall k : 1 \ldots 5$

    #define: $a_k^{IPR}, b_k^{IPR}, c_k^{IPR}\ \forall k : 1 \ldots 5$

    for $u \leftarrow 1$ **to** $u = 5$ **do** ▷ Fuzzification process

        $AO_u \leftarrow max(min(\frac{oldness_r - a_u^{AO}}{c_u^{AO}}, \frac{c_u^{AO} - oldness_r}{c_u^{AO} - b_u^{AO}}), 0)$

        $AP_u \leftarrow max(min(\frac{amount_r - a_u^{AP}}{c_u^{AP}}, \frac{c_u^{AP} - amount_r}{c_u^{AP} - b_u^{AP}}), 0)$

    end for

▷ If-Then Mamdani Rules

    $ipr_1 \leftarrow min(AO_1, AP_1)$

    $ipr_2 \leftarrow min(AO_2, AP_1)$

    $ipr_3 \leftarrow min(AO_3, AP_1)$

    $ipr_4 \leftarrow min(AO_4, AP_1)$

    $ipr_5 \leftarrow min(AO_5, AP_1)$

    $ipr_6 \leftarrow min(AO_1, AP_2)$

    $ipr_7 \leftarrow min(AO_2, AP_2)$

    $ipr_8 \leftarrow min(AO_3, AP_2)$

    $ipr_9 \leftarrow min(AO_4, AP_2)$

    $ipr_{10} \leftarrow min(AO_5, AP_2)$

    $ipr_{11} \leftarrow min(AO_1, AP_3)$

    $ipr_{12} \leftarrow min(AO_2, AP_3)$

    $ipr_{13} \leftarrow min(AO_3, AP_3)$

    $ipr_{14} \leftarrow min(AO_4, AP_3)$

    $ipr_{15} \leftarrow min(AO_5, AP_3)$

    $ipr_{16} \leftarrow min(AO_1, AP_3)$

    $ipr_{17} \leftarrow min(AO_2, AP_4)$

    $ipr_{18} \leftarrow min(AO_3, AP_4)$

    $ipr_{19} \leftarrow min(AO_4, AP_4)$

    $ipr_{20} \leftarrow min(AO_5, AP_4)$

    $ipr_{21} \leftarrow min(AO_1, AP_5)$

    $ipr_{22} \leftarrow min(AO_2, AP_5)$

    $ipr_{23} \leftarrow min(AO_3, AP_5)$

    $ipr_{24} \leftarrow min(AO_4, AP_5)$

    $ipr_{25} \leftarrow min(AO_5, AP_5)$

Algorithm 5 **(Continue)** Estimate an intensity per region with pheromones

▷ Defuzzification process

$IPR \leftarrow (ipr_5 \cdot b_1^{IPR} + (ipr_3 + ipr_4 + ipr_9 + ipr_{10} + ipr_{15} + ipr_{20}) \cdot b_2^{IPR}$

$+(ipr_1 + ipr_2 + ipr_7 + ipr_8 + ipr_{13} + ipr_{19} + ipr_{20} + ipr_{24} + ipr_{25}) \cdot b_3^{IPR}$

$+(ipr_6 + ipr_{11} + ipr_{12} + ipr_{16} + ipr_{17} + ipr_{22} + ipr_{23}) \cdot b_4^{IPR}$

$+ipr_{21} \cdot b_5^{IPR}) / \sum_{u=1}^{25} ipr_u$

return IPR

end procedure

After the intensity calculation, the regions r_{insp} that have intensity under the threshold $IPR < threshold$ to the application are discarded. Algorithm 6 performs the operations to discard these regions. Started the identification procedure, the oldness per each pheromone is calculated (line 2). After this, the fuzzy inference system is applied to estimate intensity per region (line 3). At last, the intensity is compared to the *threshold* in order to obtain the regions of interest. Once applied the identification mechanism, the MDFs must take samples of the regions of interest set ROI (line 9).

Algorithm 6 Identify the regions of interest

Input variables:

Global step in which the last MDF landed: int $globalTime$

Local variables:

Region r with pheromones: Region r_{insp}

Set of the regions with pheromones: List<Region>R_{insp}

Threshold to consider a region as valuable: double $threshold$

Set of regions considered as regions of interest: List<Region>ROI

1: **procedure** IDENTIFICATION($globalTime$)

2: CalculateOldness($globalTime$) ▷ Function to calculate the oldness of each secreted pheromone

3: CalculateIntensity() ▷ Function to calculate the intensity of each region with pheromones

4: **for all** $r_{insp} \in R_{insp}$ **do**

5: **if** $r_{insp}.intensity > threshold$ **then**

6: $ROI \leftarrow ROI \cup r_{insp}$

7: **end if**

8: **end for**

9: Harvesting() ▷ Start the harvesting stage

10: **end procedure**

Algorithm 6 **(Continue)** Identify the regions of interest

Calculate the intensity of the pheromones in a region

`Input variables:`

`Output variables:`

`Local variables:`

`- The current region being checked: Region` r

function CalculateIntensity( )
  for all $r \in R$ **do**
    if $r.Oldness \neq \oslash$ **then**
      $r_{insp}.old \leftarrow oldAverage(r)$ ▷ Calculate the oldness average per region
      $r_{insp}.amount \leftarrow countPhero(r)$ ▷ Save the number of secreted pheromones in a region
      $R_{insp} \leftarrow R_{insp} \cup r_{insp}$
    end if
  end for
  for all $r_{insp} \in R_{insp}$ **do** ▷ Calculate the intensity per each region of inspection
    $r_{insp}.intensity \leftarrow CalculateIPR(r_{insp}.old, r_{insp}.amount)$
  end for
end function

5.4 Data harvesting

The final mechanism for collaborative data-foraging is the gathering of data from previously selected regions as regions of interest. The regions of inspection that were not discarded belong to regions of interest set ROI. Each $roi \in ROI$ is assigned to an mdf to perform the data-harvesting.

The assignment of roi is according to an upward order of intensity estimated in the identification process, meaning that the first regions to be assigned are rois with less intensity. That is because the intensity in these is nearest to decay and, thus, the region of interest could disappear. If $|MDF| < |ROI|$ then the MDFs must perform various trips to accomplish the harvesting.

Algorithm 7 shows the procedure to assign the regions of interest to the MDFs. If there are available MDFs, then a roi is assigned to each mdf (line 2 and 3). Due to base station knows the regions of interest, it shares with a mdf the location of its assigned roi(line 6 and 11). After that, mdf can takeoff and sample the region.

Algorithm 7 Data harvesting of the ROIs

Input variables:

Output variables:

Local variables:

Set of MDFs: List<MDF>V_{mdfs}

Set of regions of interest: List<Region>ROI

Region of interest: Region roi

Identifier of harvester MDF: int $harvester$

Flag to indicate last harvester MDF to take off: boolean $lastH$

```
procedure HARVESTING( )
    if V_mdfs ≠ ∅ then                                        ▷ If there are available MDFs
        if ROI ≠ ∅ then                                       ▷ If there are ROIs to assign
            if ROI.length − 1 > 0 then
                harvester ← V_mdf[0]
                roi ← ROI[0]                                  ▷ Assigning a region of interest
                SendToMDF(harvester, HARVESTING, roi)  ▷ Sending a harvesting command to an MDF
                V_mdf ← V_mdf − harvester
            else
                if lastH = false then
                    harvester ← V_mdf[0]
                    roi ← ROI[0]
                    SendToMDF(harvester, HARVESTING, roi)  ▷ Sending a harvesting command to an MDF
                    V_mdf ← V_mdf − harvester
                    lastH ← true
                end if
            end if
        else
            if V_mdfs.length = mdfs then                      ▷ All mdfs have arrived
                Finish()                                      ▷ Finishing the harvesting data
            end if
        end if
    end if
end procedure
```

5.4.1 Harvester mobility

Once a region of interest has been assigned, the data should be obtained from it. To ensure that the regions are visited before the intensity of the pheromones disappears, the MDF does not perform a trip of reconnaissance again. The trail of pheromones storage regions which trace the path to return to bs from roi. It is obvious that secreted pheromones by different MDF could have different trails (some longer than another). Therefore, to visit the rois before the intensity of the pheromones disappears, the trail V_{trail_f} is used.

The MDF selects the shorter trail V_{trail_f} to return bs. However if $|V_{trail_f}| > \frac{|S_{oa}|+2}{2}$ then it means that the trip of harvesting is longer than the trip of reconnaissance. Only, in this case, the mdf should change the shorter V_{trail_f} for its counterpart V_{trail_b}. Algorithm 8 has the operations to the chosen of the shorter V_{trail}. From line 7 to 13, for each roi, the trails to return to bs are compared to them to determine which is the shorter trail. In lines 15 and 16, the change of trail is determined if V_{trail_f} is longer.

Algorithm 8 Select the short trail to achieve a ROI

Input variables:

Region of interest: Region roi

Output variables:

Set of regions to visit before to reach a ROI: $\text{List}<\text{Region}>trail$

Local variables:

Set of trails before to reach a region with food: $\text{List}<\text{trail}>BT_{roi}$

Set of trails after to reach a region with food: $\text{List}<\text{trail}>FT_{roi}$

Set of regions of interest: $\text{List}<\text{Region}>ROI$

The minimum length of a trail: int $short$

The current forward trail being checked: $\text{List}<\text{Region}>ft$

Auxiliary variable to indicate the index of a list item: int u

Index of the short trail: int $index$

1: **procedure** TrailHarvesting(roi)
2: $\quad$ $\text{List}<\text{trail}>FT_{roi} \leftarrow roi.FrontTrail$ $\quad$ ▷ All front trails in a ROI
3: $\quad$ $\text{List}<\text{trail}>BT_{roi} \leftarrow roi.BackTrail$ $\quad$ ▷ All back trails in a ROI
4: $\quad$ int $index \leftarrow 0$
5: $\quad$ int $u \leftarrow 0$
6: $\quad$ int $short \leftarrow \infty$
7: $\quad$ **for all** $ft \in FT_{roi}$ **do**
8: $\quad\quad$ **if** $ft.lenght < short$ **then**
9: $\quad\quad\quad$ $short \leftarrow ft.length$ $\quad$ ▷ Select the shortest trail
10: $\quad\quad\quad$ $index \leftarrow u$
11: $\quad\quad$ **end if**
12: $\quad\quad$ $u \leftarrow u + 1$
13: $\quad$ **end for**

Algorithm 8 **(Continue)** Select the short trail to achieve a ROI

$trail \leftarrow FT_{roi}[index]$
if $trail > BT_{roi}[index].lenght$ **then**
    $trail \leftarrow BT_{roi}[index]$ ▷ If the chosen trail is very long, then choose its backward trail
end if
return $trail$
end procedure

At the last part of the mechanism, the *mdf* follows the chosen trail. Algorithm 9 shows the travel over the shorter V_{trail}, where the *mdf* arrives at the *roi* gathers the data and back to *bs* over the same trail. In line 6 the round trip path is built, from *bs* to *roi*. While lines 7 to 12 indicate to *mdf* the next region of its trip.

Algorithm 9 Mobility for harvesting
Input variables:
Region of interest: Region r_{roi}
List of regions to visit before to reach a ROI: List<Region>*trail*
Output variables:
Local variables:
Set of regions to visit (including the ROI): List<Region>*map*
The next region to visit: Region *next*
The observation in a ROI: *package*

1: **procedure** HarvestMobility(r_{roi}, $trail$)
2: List<Region>$map \leftarrow \oslash$
3: **for** $u \leftarrow trail.length$ **to** $u \leftarrow 1$ **do** ▷ Build the way from base station to ROI
4: $map \leftarrow map \cup trail[u]$
5: **end for**
6: $map \leftarrow maps \cup r_{roi} \cup trail$ ▷ Build the way back and forth
7: **do**
8: $next \leftarrow map[0]$
9: **if** $next = r_{roi}$ **then**
10: $package \leftarrow$ TakeSample() ▷ Take an observation of the ROI
11: **end if**
12: **while** $map \neq \oslash$
13: **end procedure**

The last strategy takes advantage of the view for the MDFs instead of to require a new mobility heuristic. Data-harvesting is the last part of the proposed mechanism, with this the tasks of the Distributed Data Foraging (DDF) problem are tackled.

5.5 Experiments and results

To prove the feasibility of the indirect spatio-temporal communication protocol, this section presents the simulation results of the collaborative data-foraging mechanism. The simulations were carried out with the OMNET++ discrete simulator [Ltd, 2017b] version 5.0, along with the INET framework [Ltd, 2017a] for wireless and mobile networks, in a machine with the following characteristics:

1. OpenSUSE Leap 42.1 operative system.

2. Intel®Core™i7 processor 2.40 GHz.

3. A memory of 8 Gb.

5.5.1 Experimental setup

For the simulations, an Operational Area (OA) was modeled as a set of trajectories. Each of these trajectories is defined by an array of coordinates (x, y), where x and y denotes the center of a region $r_u \in OA$. The size of each trajectory depends on the hextille's degree related with an OA. Hence, for a hextille with degree $h = 3$ the trajectory has five pairs of coordinates, for $h = 4$ the trajectory has nine pairs of coordinates and for $h = 5$ has twelve pairs of coordinates. Whichever the size of the hextille there are twelve trajectories, however, the MDF can follow only one.

The MDFs were modeled as a mobile wireless transceiver equipped with an 802.11.x antenna. The MDFs only can communicate with base station.

5.5.2 Estimated steps for reconnaissance

The objective of this experiment is to measure the total number of steps in which the MDFs accomplish the reconnaissance task of each OA. Due to the movement of the MDFs is controlled, an MDF has $steps = \frac{h^2-h+4}{2}$ to traverse a $S_{oa} \in OA_h$. Besides, the number of needed trips for reconnaissance is 12.

Figure 5.1 shows the results of this experiment. Each colored line represents the arrival step of the last MDF at the base station. When the number of MDFs increases, the number of steps decreases. However, as from 6 MDFs, the total steps reduces in a not significant manner.

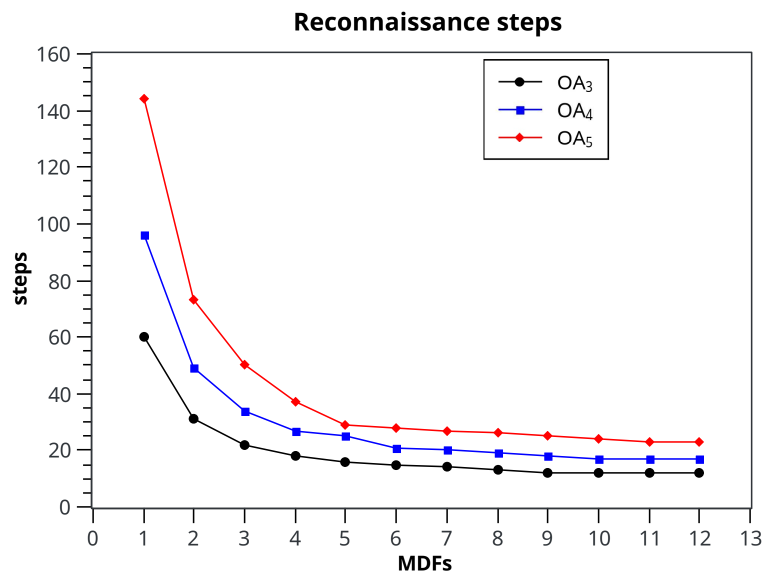


Figure 5.1: Total steps to traverse OA_h.

5.5.3 Steps to visit each region

In the above section, the total steps to traverse a whole OA_h is shown. However, the strategy of subareas S_{oa} causes an overlapping of regions; therefore, the required steps to visit each region once may vary. The average of steps to visit at least once each region of OA_h is measured in this experiment. For each OA_h, 2000 simulations were carried out, each one with a different number of MDFs.

Tables 5.1, 5.2 and 5.3 show the results for OA_3, OA_4 and OA_5 respectively. In this case, the number of steps is reduced up to 45%. If the number of MDFs is increased by one, the number of steps is reduced up to 28%, and up to 17% more

#MDFs	AVERAGE	ST DESV	MIN	MAX
1	47	5.555	30	55
2	26	2.972	16	30
3	19	2.032	12	21
4	16	1.426	11	17
5	14	1.123	10	15
6	13	1.092	10	14
7	12	0.809	10	13
8	12	0.446	10	12
9	12	0.448	10	12
10	12	0.449	10	12
11	12	0.438	10	12
12	12	0.447	10	12

Table 5.1: Average steps to visit each region in OA_3.

if $|MDF| = 3$. Similarly to the behavior of Figure 5.1, Figure 5.2 shows that if the number of MDFs is increased then the number of steps decreases.

5.5.4 Reconnaissance against data-MULEs exploration

The MULE architecture, proposed by Shah et al. [Shah et al., 2003], exploits mobile nodes to connect a Sparse Wireless Sensor Network (SWSN). In this architecture, the data-MULE does not require *a-priori* knowledge about the location of the data sources. Besides, it considers the case with many MULEs in the system without direct communication among themselves. For these features, a fair comparison between data-MULEs and our proposed solution is possible. The implementation of data-MULE was done in the OMNET++ simulation too. The MULEs do not have direct communication to each other and only communicate with the base station when they are in the same region. The MULEs depart from the base station at the same time. Due to the random movement of the MULES,

#MDFs	AVERAGE	ST DESV	MIN	MAX
1	76	8.951	48	88
2	41	4.871	25	48
3	29	3.494	18	33
4	24	2.522	17	26
5	20	2.018	16	24
6	19	1.098	13	20
7	18	1.209	13	19
8	17	1.137	13	18
9	16	0.744	13	17
10	16	0.730	13	17
11	16	0.750	13	17
12	16	0.749	13	17

Table 5.2: Average steps to visit each region in OA_4.

#MDFs	AVERAGE	ST DESV	MIN	MAX
1	113	13.332	72	132
2	61	7.398	37	72
3	43	5.493	26	49
4	35	4.168	25	38
5	29	3.419	24	36
6	27	1.126	17	28
7	26	1.759	17	27
8	24	2.628	17	26
9	22	2.515	17	25
10	21	1.738	17	24
11	21	1.102	17	22
12	21	1.120	17	22

Table 5.3: Average steps to visit each region in OA_5.

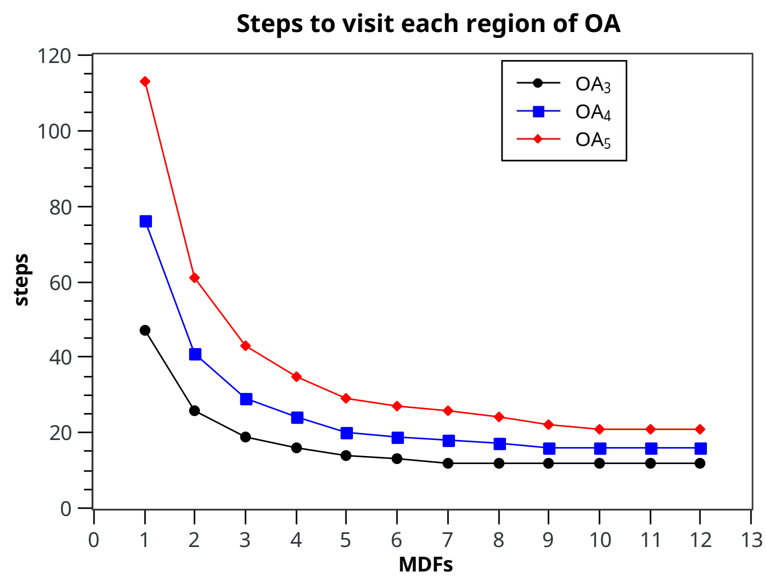


Figure 5.2: Average *steps* to visit each region of OA_h.

the simulation executed to measure the average time to reach each region in OA_h were 2000 times.

Tables 5.4, 5.5 and 5.6 show the performance of the MULE protocol in the same OA that the MDFs. Due to the randomly movement of the data-MULEs, the number of steps to reach each region in OA_h considerably increases if the OA size increases. Tables show too that even in some cases, for example in Table 5.4, the minimum number of steps to reach all regions in OA_3 with data-MULEs is minor to the number of steps taken with MDFs. However, this phenomenon requires a lucky movement of the data-MULEs during the exploration of an OA. Despite that, the standard deviation columns of Tables show a 5.528 as the minimum dispersion of the data in OA_3 with 12 data-MULEs in the system, which does not overcome the MDFs performance in the same case.

In a graphic way, Figure 5.3 shows a clear difference in the average steps taken by both protocols to reach each region at least once. Red dotted line represents the taken steps by data-MULEs, while the blue line represents the taken steps by the MDFs. In the case with a single mobile element, the reconnaissance with one MDF has a better performance than data-MULEs. The MDFs have $\approx 61\%$

#MULEs	AVERAGE	ST DESV	MIN	MAX
1	123	51.151	27	458
2	69	25.145	19	213
3	51	17.662	15	148
4	41	13.632	14	116
5	36	11.845	12	93
6	32	9.687	11	85
7	30	8.599	10	85
8	27	7.694	11	64
9	25	6.963	10	79
10	24	6.522	9	59
11	23	6.072	9	53
12	22	5.528	8	45

Table 5.4: Average steps to visit each region in OA_3.

#MULEs	AVERAGE	ST DESV	MIN	MAX
1	332	126.105	95	1269
2	188	67.317	68	563
3	135	45.080	42	373
4	112	34.605	37	309
5	97	28.735	29	217
6	85	24.804	26	206
7	78	22.600	33	207
8	71	19.788	30	168
9	66	17.832	24	157
10	62	16.473	25	143
11	59	15.531	21	148
12	56	14.063	27	138

Table 5.5: Average steps to visit each region in OA_4.

#MULEs	AVERAGE	ST DESV	MIN	MAX
1	662	241.021	214	1989
2	373	128.962	124	1146
3	276	90.134	106	767
4	224	67.269	76	496
5	192	56.168	70	527
6	170	46.908	61	419
7	153	40.739	56	355
8	143	39.097	52	336
9	131	34.066	52	298
10	123	31.872	53	283
11	118	29.913	46	244
12	112	28.397	45	251

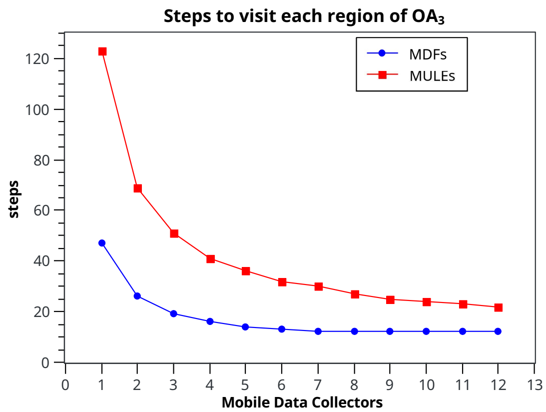
Table 5.6: Average steps to visit each region in OA_5.

fewer steps than data MULEs in OA_3 (Figure 5.3a), $\approx$ 77% in OA_4 (Figure 5.3a) and $\approx$ 82% in OA_5 (Figure 5.3a). That means if the OA_h increases, data-MULE requires more steps than MDFs to reach each region at least once.

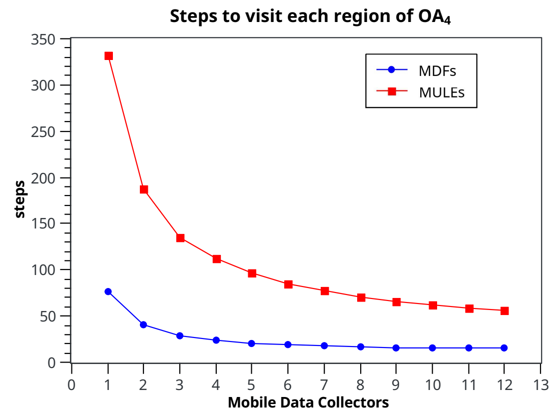
Even with the maximum number of mobiles elements (12) considered for these experiments, the difference between data-MULEs and MDFs steps is $\approx$ 45% in OA_3, $\approx$ 71% in OA_4 and $\approx$ 81% in OA_5. Therefore, in the case of multiple data-MULEs against MDFs, the reconnaissance proposed in this work has a better performance too.

5.5.5 Identification of regions of interest

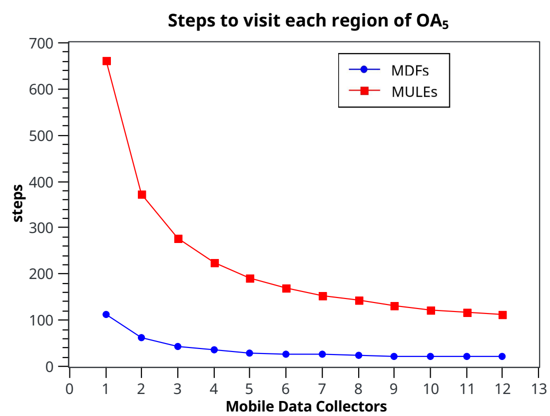
Once the reconnaissance task was tested, the next task to measure is the identification of regions of interest. The objective of this experiment is to measure the number of regions with possible profitable data sources reached before they change or disappear. The following experiment considers dynamic conditions. A



(a) Reached average steps in OA_3



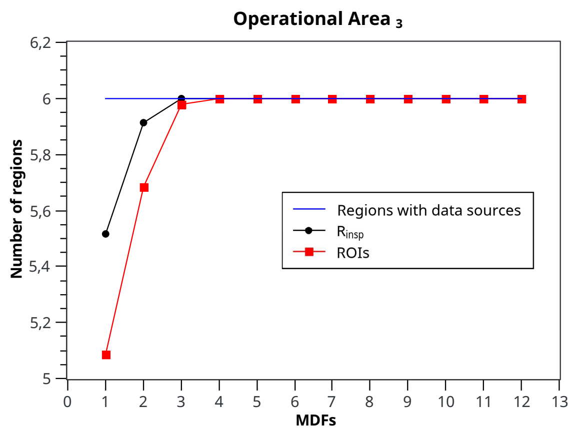
(b) Reached average steps in OA_4



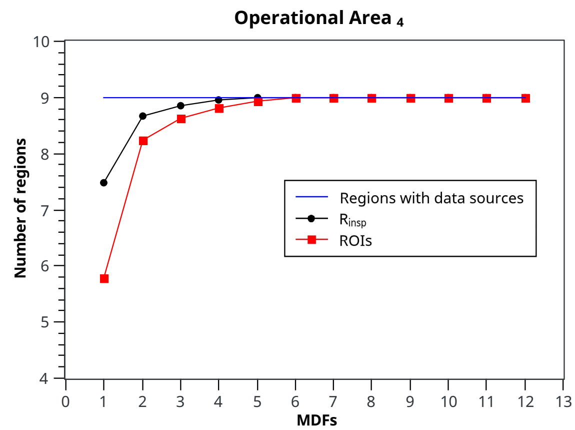
(c) Reached average steps in OA_5

Figure 5.3: Comparative between MDFs and data-MULEs

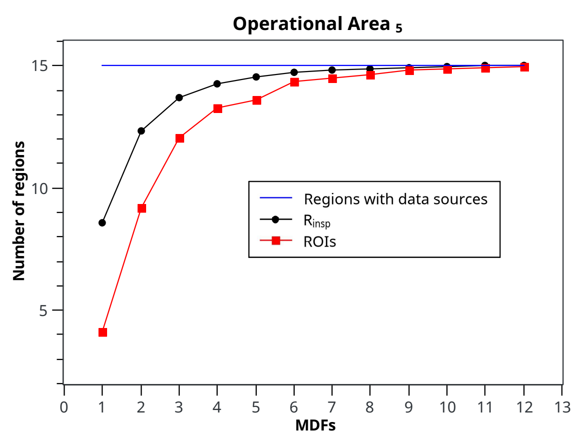
certain number of regions with profitable data sources was chosen for each OA_h, and the simulation was executed with different amount of MDFs.



(a) OA_3 with 6 regions with data source



(b) OA_4 with 9 regions with data source



(c) OA_5 with 15 regions with data source

Figure 5.4: Average regions with data source reached before it disappears

The results of the experiment are shown in Figure 5.4. The black dotted line is the regions reached and considered as possible profitable data sources, in which at least a pheromone was secreted, while the red line represents the regions of interest. The difference between amount of inspection regions $|R_{inps}|$ and regions of interest $|ROI|$ reduces when the number of MDFs increases. This difference is clearer in Figure 5.4c, OA_5 has sixty regions of which just 15 have something of value. Consequently, the reached regions at least once are minor. However, if the number of MDF increases, the reached regions increases too.

As Table 5.7 shows, when there is one MDF in the system the amount of not reached regions increases with respect to the size of the operational area. However, the number of reached regions increases as from 2 MDFs.

#MDFs	OA3	OA4	OA5	AVERAGE
1	93.016	83.144	57.166	77.775
2	98.566	96.300	82.213	92.360
3	100.00	98.266	91.466	96.577
4	100.00	99.500	94.966	98.155
5	100.00	99.922	96.846	98.922
6	100.00	100.00	98.213	99.404
7	100.00	100.00	98.746	99.582
8	100.00	100.00	99.166	99.722
9	100.00	100.00	99.613	99.871
10	100.00	100.00	99.833	99.944
11	100.00	100.00	99.966	99.988
12	100.00	100.00	99.953	99.984

Table 5.7: Percentage of regions with data sources reached

5.5.6 Comparison between identification and data-MULEs

The collaborative data-foraging mechanism can be comparable to data-MULEs at this stage. In this case, not all regions are regions of interest. Thus, it is possible

#Mobile elements	MDFs				MULEs			
	AVERAGE	ST DESV	MIN	MAX	AVERAGE	ST DESV	MIN	MAX
1	5.084	0.454	4	6	5.106	0.875	1	6
2	5.685	0.464	5	6	5.669	0.514	3	6
3	5.977	0.151	5	6	5.824	0.392	4	6
4	6.000	0.000	6	6	5.902	0.302	4	6
5	6.000	0.000	6	6	5.930	0.256	5	6
6	6.000	0.000	6	6	5.955	0.207	5	6
7	6.000	0.000	6	6	5.967	0.180	5	6
8	6.000	0.000	6	6	5.974	0.159	5	6
9	6.000	0.000	6	6	5.985	0.122	5	6
10	6.000	0.000	6	6	5.991	0.097	5	6
11	6.000	0.000	6	6	5.996	0.067	5	6
12	6.000	0.000	6	6	5.992	0.092	5	6

Table 5.8: Regions with data sources reached by MDFs and data-MULEs in OA_3

to measure the number of regions with valuable data sources obtained with MDFs against multiple data-MULEs. For this experiment, the number of regions with data sources is fixed for each OA_h, however, these regions can disappear over time. Both algorithms maintain their mobility strategies for their respective mobile elements.

In Tables 5.8, 5.9 and 5.10 a comparative between the number of regions with data sources, reached by MDFs and data-MULEs, before they disappear is shown. Table 5.8 shows a similar average number of reached regions between both solutions. Nevertheless, if the OA size increases, the number of reached regions with data sources by data-MULEs decreases, as shown Tables 5.9 and 5.10. Although for the three scenarios (OA_3, OA_4 and OA_5), the solutions using a single mobile element could have similar results, from two MDFs the difference is clear against data-MULEs. It is observable that the suitable number of MDFs in the system to perform the identification task is six.

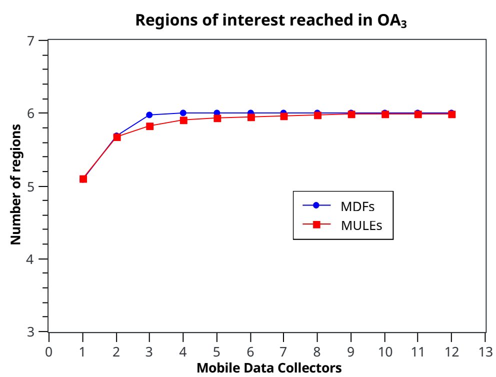
#Mobile elements	MDFs				MULEs			
	AVERAGE	**ST DESV**	**MIN**	**MAX**	**AVERAGE**	**ST DESV**	**MIN**	**MAX**
1	5.790	1.112	2	9	5.599	1.511	1	9
2	8.246	0.577	7	9	7.218	1.237	3	9
3	8.628	0.484	8	9	7.900	0.975	3	9
4	8.803	0.398	8	9	8.258	0.837	4	9
5	8.942	0.234	8	9	8.492	0.713	5	9
6	9.000	0.000	9	9	8.608	0.618	6	9
7	9.000	0.000	9	9	8.696	0.542	6	9
8	9.000	0.000	9	9	8.771	0.481	6	9
9	9.000	0.000	9	9	8.817	0.428	5	9
10	9.000	0.000	9	9	8.861	0.369	7	9
11	9.000	0.000	9	9	8.870	0.357	7	9
12	9.000	0.000	9	9	8.906	0.307	7	9

Table 5.9: Regions with data sources reached by MDFs and data-MULEs in OA_4

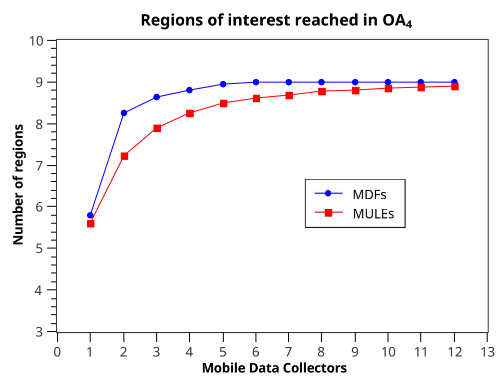
#Mobile elements	MDFs				MULEs			
	AVERAGE	**ST DESV**	**MIN**	**MAX**	**AVERAGE**	**ST DESV**	**MIN**	**MAX**
1	4.128	1.334	1	9	5.889	1.571	1	11
2	9.206	1.109	5	13	8.759	1.643	3	13
3	12.065	0.915	10	15	10.265	1.542	5	15
4	13.269	0.744	11	15	11.207	1.510	6	15
5	13.620	0.708	12	15	11.805	1.409	7	15
6	14.352	0.626	12	15	12.313	1.308	8	15
7	14.503	0.580	12	15	12.622	1.244	8	15
8	14.653	0.518	13	15	12.915	1.178	8	15
9	14.822	0.395	13	15	13.187	1.150	8	15
10	14.893	0.312	13	15	13.333	1.084	9	15
11	14.928	0.263	13	15	13.488	1.045	10	15
12	14.959	0.198	14	15	13.588	1.030	9	15

Table 5.10: Regions with data sources reached by MDFs and data-MULEs in OA_5

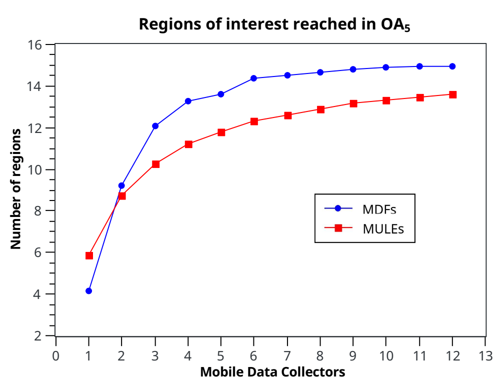
Figure 5.5 shows the average results of the experiment, using a blue dotted line to MDFs results and ed dotted line for the data-MULEs results. In the smallest scenario used for this test (OA_3), which is shown in Figure 5.5a, the number of regions reached by both algorithms looks quite similar. Graphically, in Figures 5.5b and 5.5c, the distance between lines blue and red of Figures begin to be clear if the OA size increase. In other words, the performance of collaborative data-foraging mechanism is capable of reaching more identified data sources, before those disappear, than data-MULEs.



(a) OA_3 with 6 regions with data source



(b) OA_4 with 9 regions with data source



(c) OA_5 with 15 regions with data source

Figure 5.5: Average regions with data source reached before disappearance.

5.5.7 Data harvesting

To test the final task of the collaborative data-foraging, data harvesting, the needed steps to accomplish the data-foraging are measured. The simulation was executed 2000 times to get the average per each OA_h and a fixed number of regions with data-sources was set. Just the steps for data-harvesting are counted.

Figure 5.6 shows the average steps to accomplish the data-harvesting. According to the results, if the number of MDFs in the system increases, the steps to accomplish the tasks reduce. Nevertheless, the average steps are minor to the number of regions in S_{oa}, even if the task is performed by one MDF. For example, for $|OA_4| = 36$ its average steps is 22.

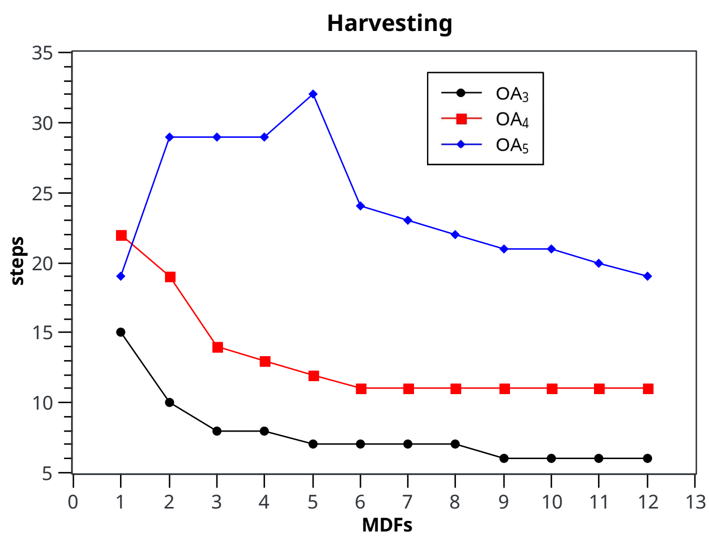
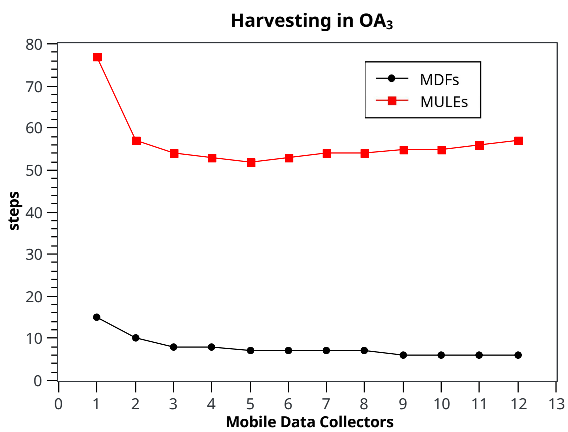


Figure 5.6: Average steps to accomplish harvesting of data

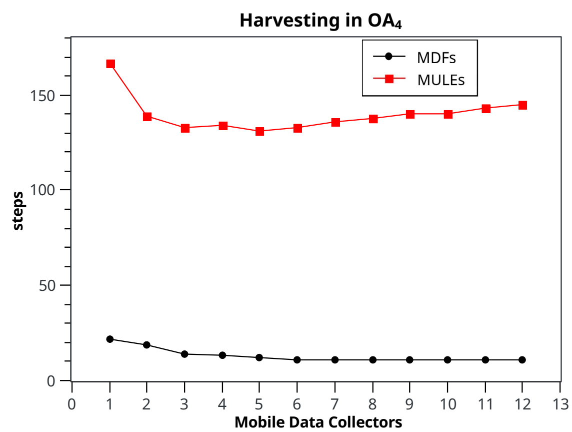
5.5.8 Data-harvesting task against data-collection

The number of steps taken only for the harvesting task in order to compare to data-MULEs is measured. The objective of this experiment is testing the performance of each algorithm. Both simulations were executed 2000 times with different amount of MDFs in OA_3, OA_4 and OA_5. The same regions of interest (ROIs) were considered for each algorithm. Besides, for data-collection, the simulation ends when each ROI is reached at least once. For this experiment, data-MULES continue with a random-walk mobility and the MDFs change to data-harvesting strategy.

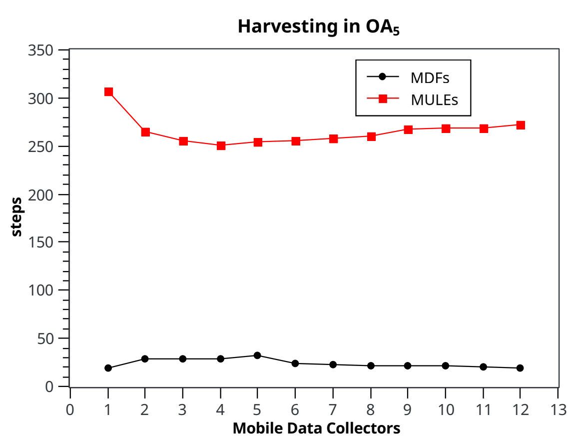
In Figure 5.7 a clear difference between the steps taken by data-harvesting performed by MDFs (black line) and data-collection performed by data-MULEs (red line) is shown. The steps in data-collection performed by MULEs increases in a significant manner if the OA_h size increment, while the selection of the regions and the pheromones trails by the MDFs reduce these steps.



(a) Average steps to reach the ROIs in OA_3



(b) Average steps to reach the ROIs in OA_4



(c) Average steps to reach the ROIs in OA_5

Figure 5.7: Comparison between data-harvesting and data-collection

#Mobile elements	MDFs				MULEs			
	AVERAGE	**ST DESV**	**MIN**	**MAX**	**AVERAGE**	**ST DESV**	**MIN**	**MAX**
1	15	1.565	12	18	77	43.058	6	351
2	10	0.931	8	11	57	25.628	8	203
3	8	0.295	6	9	54	22.675	12	160
4	8	0.213	7	8	53	20.862	14	174
5	7	0.350	6	8	52	18.986	16	145
6	7	0.282	6	7	53	18.559	16	158
7	7	0.205	6	7	54	18.13	16	144
8	7	0.112	6	7	54	18.338	17	158
9	6	0.000	6	6	55	18.212	17	149
10	6	0.000	6	6	55	17.467	19	143
11	6	0.000	6	6	56	18.099	18	159
12	6	0.000	6	6	57	18.139	19	159

Table 5.11: Data-harvesting against data-collection in OA_3

In addition to the depicted by Figure 5.7, Tables 5.11, 5.12 and 5.13 display statistical evidence of the results. The standard deviation (**ST DESV** column) for data-MULEs is greater than MDFs due to their randomly mobility strategy. Besides, for data-MULEs, that standard deviation increases significantly (more than 200%) according to the OA size. On the other side, the MDFs do not need to visit each region in OA. Thus, the standard deviation for data-harvesting does not increase like data-collection. However, in operational areas with a significant amount of regions increases the number steps of trails to reach the regions of interest.

#Mobile elements	MDFs				MULEs			
	AVERAGE	ST DESV	MIN	MAX	AVERAGE	ST DESV	MIN	MAX
1	22	5.721	4	38	167	96.220	5	845
2	19	1.632	14	23	139	67.260	19	528
3	14	1.210	12	18	133	54.533	23	464
4	13	1.198	10	16	134	51.762	21	398
5	12	1.314	9	15	131	49.443	33	459
6	11	1.098	9	14	133	48.904	43	398
7	11	1.206	8	14	136	47.565	37	394
8	11	1.106	9	14	138	46.558	40	366
9	11	2.244	9	30	140	46.907	46	353
10	11	1.019	8	14	140	44.99	42	377
11	11	0.995	9	14	143	45.822	47	390
12	11	1.034	8	13	145	44.076	54	355

Table 5.12: Data-harvesting against data-collection in OA_4

#Mobile elements	MDFs				MULEs			
	AVERAGE	ST DESV	MIN	MAX	AVERAGE	ST DESV	MIN	MAX
1	19	9.165	2	58	307	175.991	12	1196
2	29	4.828	14	49	265	119.134	34	802
3	29	3.069	19	38	256	107.28	54	899
4	29	2.354	21	37	251	97.757	57	749
5	32	2.119	25	39	254	95.798	70	829
6	24	1.921	17	30	256	95.466	62	796
7	23	1.667	17	28	258	93.149	75	851
8	22	1.605	17	26	261	93.256	80	871
9	21	1.647	16	27	267	91.416	86	821
10	21	1.498	16	25	269	90.741	85	761
11	20	1.347	16	24	269	87.738	82	791
12	19	1.268	15	22	272	88.351	90	684

Table 5.13: Data-harvesting against data-collection in OA_5

5.5.9 Discussion

In this section, several experiments were presented to prove the feasibility of the proposed protocol and mechanism. Based on the results obtained from each experiment, it is stated that the increment of the number of MDFs in the system decreases the number of steps to perform any task of the data foraging problem for any size of the operational area. Besides, it is possible to determine a suitable amount of MDFs, which is between four and six. For the reconnaissance mechanism, more than six MDFs do not represent a significant gain in the reduction of the number of steps.

Due to the dynamical characteristics of the environment, the unknowing of the data sources' location, the features of a Sparse Mobile Wireless Sensor Network and the use of multiple mobile data collectors with a limited range communication; the majority of the related work cannot be directly compared with the collaborative data-foraging mechanism.

However, some MULE architecture [Shah et al., 2003] features allow to make a comparative between data-collection with data-MULEs and data-foraging with MDFs. The MULE architecture does not require *a-priori* knowledge about the location of the data sources and, in a scheme with multiple elements, the data-MULEs do not share information among themselves. The objective of the MULE is to pick up the data of the sensors and carry them until found the base station. So, the data-MULE delivers the retrieved data to the base station and continues with its task. Therefore, each stage of collaborative data-foraging was compared with this work.

The results of each stage of collaborative data-foraging mechanism against data-collection using MULEs show the following:

- The reconnaissance with MDFs has a better performance than data-collection with MULEs. In the reconnaissance, the number of steps required to visit each region at least once depends on the number of Mobile Data Collectors

(MDCs) in the system. A greater amount of MDFs or data-MULEs reduce the number steps to visit each region. However, the results show a clear difference among the required steps by data-MULEs and MDFs, being the first ones whose require a higher number of steps than the second ones.

- During the identification task, a similar number of reached regions of interest by both solutions is observable. However, the standard deviation of the data-MULEs performance is greater if the size of the OA increases. In that case, the number of regions, reached by data-MULEs, is considerably reduced.

- For the final task, data harvesting, an MDF takes advantage of other MDFs' views. During the reconnaissance trip and the release of pheromones, the trails to return to base station were recorded. The MDFs follow the shorter trails to reach the regions of interest while data-MULEs are still with random-walk mobility. Thus, the needed steps to accomplish this task is reduced.

From the simulations, it can be observed that the collaborative mechanism has a better performance than the data collection with data-MULEs, in the same conditions for any of the three tasks. However, the collaborative data-foraging has some disadvantages over data-MULEs, which are described as follows:

- The base station must build the sub operational areas before to begin the data-foraging. Unlike data-MULEs, the collaborative data-foraging requires a previous process to divide a sensing area into sub-areas and assign it to an MDF. Once assigned the sub-area, the MDF starts with the first task of data-foraging.

- The collaborative data-foraging mechanism requires processing in the base station to determine if a region is a region of interest. In data-MULE architecture, the data-MULEs just pick up the data, and eventually, the data is delivered to the base station. It means that, for data-MULEs, one observation is sufficient to discrete a region as a region of interest.

5.6 Performance analysis

The performance, of the spatio-temporal communication protocol and of the collaborative data-foraging mechanism, is determined by analyzing their overhead in terms of resources' requirements.

5.6.1 Storage overhead and computational cost

The hexagonal tessellation used to represent the sensing area can be viewed as a regions' vector. For the collaborative data-foraging, a regions' vector of size n is given before to begin the data-foraging tasks. For the reconnaissance task twelve S_{oa} with $s = (n-1)/6$ each one are built. To identity regions of interest, the mechanism uses artificial pheromones with the data structures r_u, mdf_v, ts, V_{trail_b}, and $Vtrail_f$. The number of pheromones depends directly on n. Due to the overlapping of S_{oa}, during identification task, the number of pheromones released in the regions is two except in the nearest regions to the base station, where the maximum number is four. The maximum amount of pheromones storage in a vector of size equals n is $2n + 10$.

Let b the number of bytes used to represent an integer value and k the number of pheromones secreted in a region u, the storage overhead of a region is determined by Equation 5.1:

$$SC_f = k * b(|r_u| + |mdf_v| + |ts| + |V_{trail_b}| + |V_{trail_f}|) \quad (5.1)$$

Assuming the worst case, where all regions in a sensing area have pheromones, as indicated in Equation 5.2:

$$SC_f = b(2n + 10) \quad (5.2)$$

Therefore, by 5.1 the asymptotic storage overhead is linear: $SC_f \sim \mathcal{O}(n)$.

The computational cost of the collaborative data-foraging mechanism is measure through the number of steps to traverse a whole sensing area. Although there are $n-1$ regions to visit, an MDF requires $s+2$ steps to traverse one sub operational area. Considering twelve sub operational arean S_{oa} in a sensing area, the number of steps to traverse a sensing area is determined by Equation 5.3:

$$CC = 12(s+2) = 2n + 22 \tag{5.3}$$

Therefore, by 5.3 the asymptotic computational cost is indicated by Equation 5.4:

$$CC \sim \mathcal{O}(n) \tag{5.4}$$

Chapter 6

Conclusions and future work

6.1 Summary

This thesis has presented an indirect spatio-temporal communication protocol for Sparse Mobile Wireless Sensor Network (SMWSN). The protocol was designed to allow the information exchange among entities without enduring transmission links. This capability is useful in sensing dynamic environments where the data sources can change or disappear. The indirect spatio-temporal communication protocol is inspired by the stigmergy principle of secretion of pheromones. Pheromones were modeled as an abstract data type whose operations allow the establishment of fuzzy-causal dependencies among exchanged messages. Such dependencies permit to partially arrange messages considering where and when they were retrieved. In this way, the usefulness of the exchanged information makes sense considering a spatio-temporal timeline.

The proposed protocol is the core of the collaborative data-foraging mechanism presented in Chapter 5. Through a group of mobile data foragers (MDF) the mechanism collectively performs reconnaissance, identifies and retrieves profitable data from non-perdurable sources, without enduring transmission links not requiring synchronized physical clocks; satisfying the features and constraints of an SMWSN. One original aspect of the mechanism is that to find profitable data sources, each MDF uses a travel heuristic that does not require the *a-priori* knowledge of the data sources locations neither from the whole environment.

6.2 Conclusions

The feasibility of the mechanism was proved through simulations. In such experiments, the performance of the proposed mechanism was evaluated considering the number of steps required to complete the data-foraging process. Simulation results show that the proposed mechanism reduces the number of steps until 84% compared to the data-MULEs approach, which is the nearest solution for data-foraging.

Moreover, for each task of data-foraging (reconnaissance, identification of ROIs and data-harvesting), the collaborative data-foraging had better performance than a multiple data-MULE approach. Despite similar results in identification task obtained by both protocols (for example in an OA_3), the number of steps needs to reach these regions by MULE is higher than the required steps by the collaborative data-foraging mechanism.

In addition, the results show that the collaborative foraging has better performance than the single foraging. In the case of two MDF the gain is up to 45%, in the case of three MDF the gain is up to 59%, in the case of four MDFs the gain is up to 66%, in the case of five MDFs the gain is up to 70%, while in the case of six MDFs the gain is up to 71%. The evidence shows that the employment of more than six MDFs has no significant gain.

6.3 Future work

Some aspects, observed during the development of the proposed protocol and mechanism, deserve further studies.

- **Increase the efficiency of the data-foraging mechanism**. In the current mechanism all the MDFs perform the same task at the same time, i.e. the three collaborative tasks of data-foraging are sequentially executed. The MDFs could perform these three tasks in a parallel manner in order to take advantage of the waiting time of some MDFs to overtake other tasks. This could allow to some MDFs to identify new data sources if these appear, while other MDFs perform harvesting.

- **Collaborative direct and indirect communication among MDFs**. In this work the collaborative data-foraging problem based on indirect communication was tackled. However, under this scheme, the MDFs cannot directly communicate among themselves during a trip. This condition restricts the MDF to wait until reaching the base station to exchange the retrieved data. Thereby, a future extension of the data-foraging mechanism is related to incorporate direct communication capabilities among MDFs, which open two main challenges:
  - **Discovery and data transfer**, which includes coordination mechanisms to take advantage of the contact area created when an MDF is within the transmission range of other MDFs.
  - **Data alignment mechanisms for the retrieval of multiple local views**, which includes data alignment strategies to tackle the problem of inconsistencies among concurrent samples reported by different sources.
- **Pheromones trail based on the quality of the collected data**. In the indirect spatio-temporal communication protocol proposed, the pheromones only consider the amount and oldness of the observations without taking account the quality of the samples retrieved in the region. Nevertheless, for some applications, in addition to the spatio-temporal context, the quality of the data is important. Therefore, another extension of the work is to incorporate context-aware information to enrich the semantic of pheromone trails.

Bibliography

[Beckers et al., 1989] Beckers, R., Goss, S., Deneubourg, J.-L., and Pasteels, J.-M. (1989). Colony size, communication, and ant foraging strategy. *Psyche*, 96(3-4):239–256.

[Betke et al., 1995] Betke, M., Rivest, R. L., and Singh, M. (1995). Piecemeal learning of an unknown environment. *Machine Learning*, 18(2-3):231–254.

[Castañeda Cisneros et al., 2017] Castañeda Cisneros, J., Pomares Hernandez, S. E., Perez Cruz, J. R., Rodríguez-Henríquez, L. M., and Gonzalez Bernal, J. A. (2017). Data-foraging-oriented reconnaissance based on bio-inspired indirect communication for aerial vehicles. *Applied Sciences*, 7(7):729.

[Chakrabarti et al., 2003] Chakrabarti, A., Sabharwal, A., and Aazhang, B. (2003). Using predictable observer mobility for power efficient design of sensor networks. In *Information Processing in Sensor Networks*, pages 129–145. Springer.

[Chandra and Kshemkalyani, 2005] Chandra, P. and Kshemkalyani, A. D. (2005). Causality-based predicate detection across space and time. *IEEE Transactions on Computers*, 54(11):1438–1453.

[Chen and Xu, 2005] Chen, X. and Xu, M. (2005). A geographical cellular-like architecture for wireless sensor networks. In *International Conference on Mobile Ad-Hoc and Sensor Networks*, pages 249–258. Springer.

[Coulouris et al., 2012] Coulouris, G., Dollimore, J., and Kindberg, T. (2012). *Distributed Systems: Concepts and Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 5th edition.

[Di Francesco et al., 2011] Di Francesco, M., Das, S. K., and Anastasi, G. (2011). Data collection in wireless sensor networks with mobile elements: A survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(1):7.

[Dipple et al., 2014] Dipple, A., Raymond, K., and Docherty, M. (2014). General theory of stigmergy: Modelling stigma semantics. *Cognitive Systems Research*, 31:61–92.

[Djidjev and Potkonjak, 2012] Djidjev, H. and Potkonjak, M. (2012). Dynamic coverage problems in sensor networks. *Intelligent Sensor Networks: The Integration of Sensor Networks, Signal Processing and Machine Learning*, page 459.

[Dominguez et al., 2005] Dominguez, E. L., Ramirez, J. E., Fanchon, J., and Hernandez, S. E. P. (2005). A fault-tolerant causal broadcast algorithm to be applied to unreliable networks. In *IASTED PDCS*, pages 465–470.

[Dorigo et al., 2000] Dorigo, M., Bonabeau, E., and Theraulaz, G. (2000). Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16(8):851–871.

[Erman et al., 2012] Erman, A. T., Dilo, A., and Havinga, P. (2012). A virtual infrastructure based on honeycomb tessellation for data dissemination in multi-sink mobile wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):17.

[Evans and Karwowski, 1986] Evans, G. W. and Karwowski, W. (1986). A perspective on mathematical modeling in human factors. In *Advances in Human Factors/Ergonomics*, volume 6, pages 3–27. Elsevier.

[Faiçal et al., 2014] Faiçal, B. S., Costa, F. G., Pessin, G., Ueyama, J., Freitas, H., Colombo, A., Fini, P. H., Villas, L., Osório, F. S., Vargas, P. A., et al. (2014). The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides. *Journal of Systems Architecture*, 60(4):393–404.

[Goss et al., 1989] Goss, S., Aron, S., Deneubourg, J.-L., and Pasteels, J. (1989). Self-organized shortcuts in the argentine ant. naturwissenschaften 76: 579-581. 76:579–581.

[Grassé, 1959] Grassé, P.-P. (1959). La reconstruction du nid et les coordinations interindividuelles chezbellicositermes natalensis etcubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. *Insectes sociaux*, 6(1):41–80.

[Grunbaum and Shephard, 1977] Grunbaum, B. and Shephard, G. C. (1977). Tilings by regular polygons. *Mathematics Magazine*, 50(5):227–247.

[Gu et al., 2006] Gu, Y., Bozdağ, D., Brewer, R. W., and Ekici, E. (2006). Data harvesting with mobile elements in wireless sensor networks. *Computer Networks*, 50(17):3449–3465.

[Hernandez et al., 2004] Hernandez, S. P., Fanchon, J., and Drira, K. (2004). The immediate dependency relation: an optimal way to ensure causal group communication. *Annual Review of Scalable Computing*, 3:61–79.

[Holland and Melhuish, 1999] Holland, O. and Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. *Artificial life*, 5(2):173–202.

[Jackson and Ratnieks, 2006] Jackson, D. E. and Ratnieks, F. L. (2006). Communication in ants. *Current biology*, 16(15):R570–R574.

[Jea et al., 2005] Jea, D., Somasundara, A., and Srivastava, M. (2005). Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Distributed computing in sensor systems*, pages 244–257. Springer.

[Jia and Zhou, 2004] Jia, W. and Zhou, W. (2004). *Distributed network systems: from concepts to implementations*, volume 15. Springer Science & Business Media.

[Juang et al., 2002] Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L. S., and Rubenstein, D. (2002). Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. *ACM SIGARCH Computer Architecture News*, 30(5):96–107.

[Kramer, 2001] Kramer, D. L. (2001). Foraging behavior. *Evolutionary ecology: concepts and case studies. Oxford University Press, New York*, pages 232–246.

[Lamport, 1978] Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565.

[Lee, 1990] Lee, C.-C. (1990). Fuzzy logic in control systems: fuzzy logic controller. i. *IEEE Transactions on systems, man, and cybernetics*, 20(2):419–435.

[Lee et al., 2009] Lee, U., Magistretti, E., Gerla, M., Bellavista, P., Lió, P., and Lee, K.-W. (2009). Bio-inspired multi-agent data harvesting in a proactive urban monitoring environment. *Ad Hoc Networks*, 7(4):725–741.

[Liu et al., 2006] Liu, R. P., Rogers, G., and Zhou, S. (2006). Wsn14-3: Honeycomb architecture for energy conservation in wireless sensor networks. In *Global Telecommunications Conference, 2006. GLOBECOM'06. IEEE*, pages 1–5. IEEE.

[Ltd, 2017a] Ltd, O. (2017a). Inet framework. `https://inet.omnetpp.org/`. Accessed: 2017-08-17.

[Ltd, 2017b] Ltd, O. (2017b). Omnet++ discrete event simulator. `https://omnetpp.org/`. Accessed: 2017-08-17.

[Mamdani and Assilian, 1975] Mamdani, E. H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1):1–13.

[Mattern et al., 1989] Mattern, F. et al. (1989). Virtual time and global states of distributed systems. *Parallel and Distributed Algorithms*, 1(23):215–226.

[Munir et al., 2007] Munir, S. A., Ren, B., Jiao, W., Wang, B., Xie, D., and Ma, J. (2007). Mobile wireless sensor network: Architecture and enabling technologies for ubiquitous computing. In *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, volume 2, pages 113–120. IEEE.

[Nandi et al., 2014] Nandi, S., Panigrahy, N., Agrawal, M., Ghosh, S. C., and Das, S. (2014). Efficient channel assignment for cellular networks modeled as honeycomb grid. In *ICTCS*, pages 183–295.

[Panait and Luke, 2004] Panait, L. and Luke, S. (2004). A pheromone-based utility model for collaborative foraging. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 36–43. IEEE Computer Society.

[Perez Cruz and Pomares Hernandez, 2014] Perez Cruz, J. R. and Pomares Hernandez, S. E. (2014). Temporal data alignment and association for event-streaming in ubiquitous environments based on fuzzy-causal dependencies. In *Electronics, Communications and Computers (CONIELECOMP), 2014 International Conference on*, pages 127–134. IEEE.

[Qu et al., 2015] Qu, Y., Zhang, Y., and Zhang, Y. (2015). A uav solution of regional surveillance based on pheromones and artificial potential field theory. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pages 380–385. IEEE.

[Ross, 2009] Ross, T. J. (2009). *Fuzzy logic with engineering applications*. John Wiley & Sons.

[Ross, 2010] Ross, T. J. (2010). Properties of membership functions, fuzzification, and defuzzification. *Fuzzy Logic with Engineering Applications, Third Edition*, pages 89–116.

[Sayyed and Becker, 2015] Sayyed, A. and Becker, L. B. (2015). A survey on data collection in mobile wireless sensor networks (mwsns). In *Cooperative Robots and Sensor Networks 2015*, pages 257–278. Springer.

[Shah et al., 2003] Shah, R. C., Roy, S., Jain, S., and Brunette, W. (2003). Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(2):215–233.

[Sharieh et al., 2008] Sharieh, A., Qatawneh, M., Almobaideen, W., and Sleit, A. (2008). Hex-cell: modeling, topological properties and routing algorithm. *European Journal of Scientific Research*, 22(2):457–468.

[Somasundara et al., 2004] Somasundara, A. A., Ramamoorthy, A., and Srivastava, M. B. (2004). Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International*, pages 296–305. IEEE.

[Stranders et al., 2013] Stranders, R., De Cote, E. M., Rogers, A., and Jennings, N. R. (2013). Near-optimal continuous patrolling with teams of mobile information gathering agents. *Artificial intelligence*, 195:63–105.

[Sugeno, 1985] Sugeno, M. (1985). An introductory survey of fuzzy control. *Information sciences*, 36(1-2):59–83.

[Takagi and Sugeno, 1985] Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1):116–132.

[Van Le et al., 2014] Van Le, D., Oh, H., and Yoon, S. (2014). A novel hierarchical cooperative data gathering architecture using multiple mobile elements. In *Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on*, pages 522–527. IEEE.

[Vasilescu et al., 2005] Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., and Corke, P. (2005). Data collection, storage, and retrieval with an underwater sensor network. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 154–165. ACM.

[Yan et al., 2010] Yan, H., Huo, H., Xu, Y., and Gidlund, M. (2010). Wireless sensor network based e-health system-implementation and experimental results. *IEEE Transactions on Consumer Electronics*, 56(4):2288–2295.

[Yick et al., 2008] Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer networks*, 52(12):2292–2330.

[Yu et al., 2008] Yu, R., Wang, X., and Das, S. K. (2008). Efficient data gathering using mobile elements in partially connected sensor networks. In *Control and Decision Conference, 2008. CCDC 2008. Chinese*, pages 5337–5342. IEEE.

[Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3):338–353.

[Zadeh, 1973] Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on systems, Man, and Cybernetics*, (1):28–44.

[Zhang et al., 2009] Zhang, X., Zhang, L., Chen, G., and Zou, X. (2009). Probabilistic path selection in wireless sensor networks with controlled mobility. In *Wireless Communications & Signal Processing, 2009. WCSP 2009. International Conference on*, pages 1–5. IEEE.

[Zhao et al., 2004] Zhao, W., Ammar, M., and Zegura, E. (2004). A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 187–198. ACM.