SOLVING THE COMMUNICATION CHANNEL ASSOCIATION PROBLEM FOR MOBILE ROBOTS

A Thesis

by

PLAMEN IVANOV IVANOV

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Dylan Shell
Committee Members,	Nancy Amato
	Suman Chakravorty
Head of Department,	Dilma De Silva

August 2015

Major Subject: Computer Science

ABSTRACT

Robots working in teams can benefit from recruiting the help of conveniently located nearby robots. To do this an initiating robot needs to be aware of the network addresses of its neighbors. However, a robot is typically aware of its neighbors' relative positions through locally sensed information, such as range and bearing, which does not include the network ID of the neighbor. In this work, robots use a simple visual gesture, such as a light being turned on or off, paired with wireless messages to rapidly and effectively establish a one-to-one association between the relative positions (visual IDs) of neighboring robots and their network addresses (wireless IDs).

We identify and formalize the problem of associating the two types of IDs — the association problem, and explore its structure in detail. We also identify that the visual gesture along with the sensor which detects it form a second communication channel that is used by the robots to transmit information. We present two deterministic and one probabilistic algorithm which solve the association problem for stationary robots. Furthermore, we introduce modifications to the probabilistic algorithm in order to tackle the harder, mobile robot version of the problem where robot motion results in changing connectivity between robots. Our algorithmic approach exploits the physically situated properties of the visual IDs to help solve the association problem. We identify key parameters, such as robot density, communication range and movement speed, and study their effect on the performance of the probabilistic algorithm. We use a population growth modeling framework, called Branching Processes, as part of a set of models for the association process which can be used to predict the macroscopic performance of a multi-robot system running the probabilistic algorithms. This set of models can be used to determine how successful the probabilistic algorithm is at solving the association problem in any multi-robot system based on the above mentioned key parameters. The framework can also be used to fine-tune parameters when designing a system so that its performance achieves some desired threshold.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

1. INTRODUCTION*

Robots working in multi-robot systems often need to employ spatially coordinated communication in order to cooperate actively and deliberately. For example, a robot may want to send the request: "*Will the robot to my left help me move this piano?*" Typically such robots can identify the targeted message recipients through a sensor (*e.g.*, camera) but use a broadcast device (*e.g.*, Wi-Fi, radio, etc.) to send a message. Since the same robot will have different identifiers in these two mediums, an association between them needs to be formed in order for the request for help over the network to be meaningful to the robots which receive it. So, the example request then becomes: "*Will robot 192.168.0.56 help me move this piano?*" This situation, shown in Fig. 1.1, presents a problem for robots which have no direct knowledge of the network addresses of their neighbors: How does a robot learn the network address of a robot it wants to communicate with? We address the question of relating a local, relative view of a robot to an identifier which can be used to address the robot directly. The multi-robot system we envision is composed of anonymous robots which cannot be distinguished from one another through observation of their features. The robots in the system also have no access to a common position reference frame.

The Wi-Fi and camera form two independent communication channels through which the robots can exchange information. The camera of a robot can be converted into a visual communication channel by enabling the robots to produce a visually identifiable and agreed upon gesture (*e.g.,* an instance of a protocol on just such a visual communication channel is given in [3]). In terms defined by [17], the wireless

*Part of this chapter is reprinted with permission from P. Ivanov and D. A. Shell. Associating nearby robots to their voices. In *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALife)*, 2014, Copyright [2014] by Plamen Ivanov, Dylan Shell.

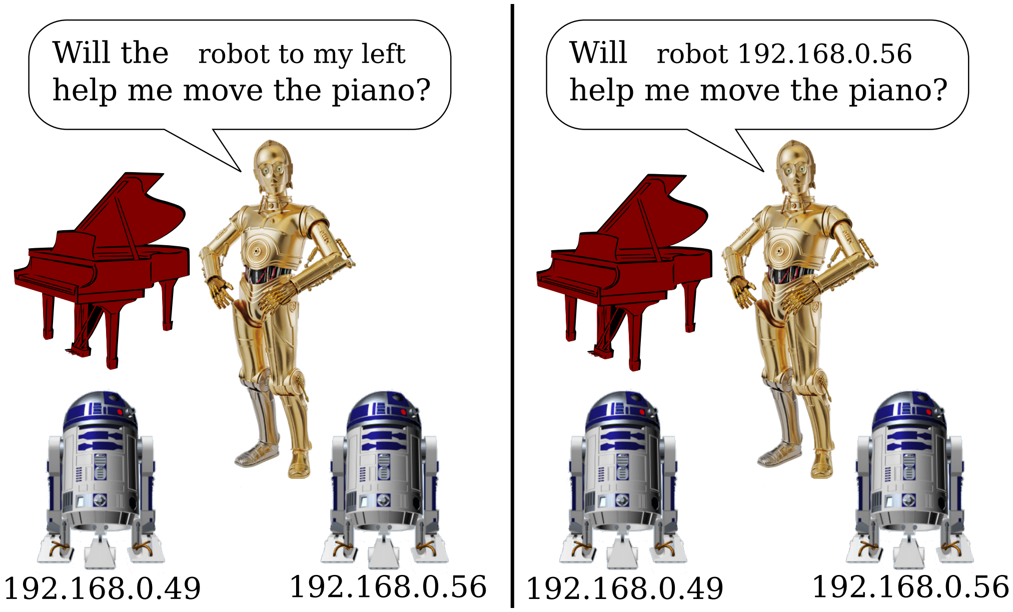

Will the robot to my left help me move the piano?
192.168.0.49
192.168.0.56
Will robot 192.168.0.56 help me move the piano?
192.168.0.49
192.168.0.56

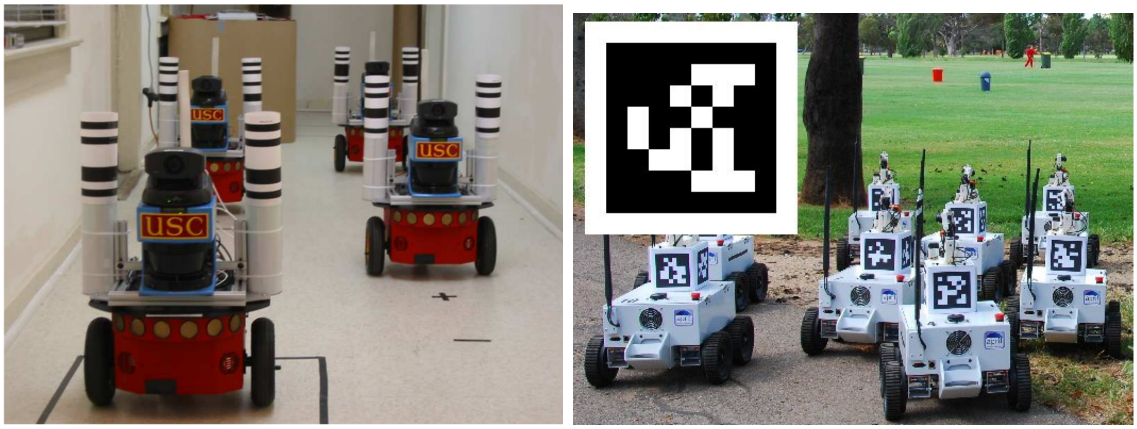
Figure 1.1: Robots which cannot distinguish one another visually run into communication problems when they need to unambiguously address specific recipients.

and visual channel are categorized as abstract and situated communication channels, respectively. Abstract channels carry meaning only in the message contents, while situated channels carry meaning in both the message and some property of the channels themselves. For the problem we are addressing the relevant property of the situated visual channel is the relative position of a neighboring robot.

With the above concepts in mind, the specific problem we are examining in this work is how, given an abstract and a situated communication channel, can all robots solve the following problem in parallel:

> *For every robot identifiable in a situated communication channel, determine and associate the robot's corresponding identifier in the abstract communication channel.*

We call this the **communication channel association problem** or **association problem** for short.



(a) Fiducial markers as described in [10] (b) AprilTag markers as described in [16]

Figure 1.2: Examples of visual markers used for identifying robots.

The typical approach to solving the association problem is by defining static lists of associations between robot network addresses and their corresponding visual identifiers in an off-line manner. A good example of this approach are the ubiquitous visual fiducial or marker systems [10, 6, 16]. An example of two such visual marker systems can be seen in Fig. 1.2.

In this work we present an algorithmic approach for solving the association problem. Our approach does not rely on global information or external markers. Instead, the robots use a simple, visually identifiable gesture, such as a light being turned on and off and wireless communication to quickly solve the association problem. The IDs of robots in the visual channel are locally assigned by each observing robot and are typically the range and bearing to the visually identified robots. The IDs of the robots in the wireless channel are assigned to the robots globally and do not depend on other robots' perspectives.

The robots running the algorithm operate in a probabilistic environment and solve the association problem locally, without depending on the system size. Consequently, we are interested in understanding and predicting the expected behavior of a large system of robots running our algorithm. For this purpose we run simulations on systems with varying parameters, such as density of robots in the environment, communication range, and robot speed and employ a predictive model which can give the expected macroscopic performance of such systems.

This work has the following contributions. We identify and formalize the communication channel association problem. We highlight key concepts underlying the problem such as information space [12], situated communication versus abstract communication [17], and indexical versus objective knowledge [1, 13]. We show that the problem can be split into two broad categories — stationary robot case, mobile robot case, and that these two categories are amenable to different analysis approaches. We present several algorithms which can be used to solve either the stationary case (deterministic and probabilistic algorithms) or the mobile case (probabilistic algorithm) of the association problem. We identify three key system parameters — robot density, communication range size, and robot velocity — and examine their effect on the algorithms' performance through simulation. We introduce a set of models which incorporate the theory of Branching Processes [9] in order to efficiently predict the macroscopic behavior of a system solving the association problem with our probabilistic algorithms. The models are validated by comparing their predictions to the simulation results. We also show that the Markov chain analysis approach [7] is very expensive when applied to the association problem as its structure is probabilistic and highly combinatorial in nature.

The remainder of the thesis is organized as follows. In Chapter 2, we examine the related work in more detail. Chapter 3 gives a formal specification of the communication channel association problem and a description of the motivating multi-robot system which we use throughout the work. Chapter 4 shows the algorithms and predictive models for the stationary robot version of the association problem. Chapter 5 shows the algorithm and models for the mobile robot version of the association problem. Finally, in Section 6 we discuss future work and conclude.

2. RELATED WORK AND BACKGROUND

2.1 Related Work

The idea of situated communication, as described in [17], is central to solving the communication channel association problem. A situated communication channel is one where a property of the channel itself gives additional relevant meaning to a message received through the channel. In our case, the additional meaning is the relative position of the transmitting robot, which allows robots to associate a meaningful locally relevant property of neighboring robots to their wireless IDs. Another, more general concept, which encompasses the situated property of the communication channel, is indexical knowledge, as described in [1, 13]. Indexical knowledges refers to knowledge, which is relative to the observer. It is contrasted by objective knowledge, which is independent from the perspective of the observer.

We informally categorize robotic systems based on the number of communication channels that are available to them. The most common cases are one-channel (*e.g.* wireless) and two-channel systems (*e.g.* infrared communication devices). In [3], the authors examine a scenario where the typical one-channel approach — wireless — is unavailable and robots can only communicate through *movement-signals*. This works is an example where an explicit communication channel can be established using robot sensors and robot gestures — in this work the gestures are composed of specific motion of a robot through the environment. Since this system has only one communication channel, the association problem is not meaningful.

A multi-robot system can explicitly fall under the two-channel category, if its robots are equipped with 2 independent communication devices such as wireless and infrared-based communication. In such systems typically both communication channels have enough bandwidth to reliably and quickly transmit a robot's ID. In addition, the infrared communication channel is also a situated one — it can be used to detect range and bearing to other robots, as in [8]. Since each channel can transmit a robot's ID, this means that the association problem can be solved by directly sending a message containing the ID. Robots which receive the message now know the robot's identity and can address them directly through either channel.

As we demonstrate in our work and as can be seen from [3], a communication channel can be established through the use of any robot sensor which can detect a change in the state of another robot. In [3], this communication channel uses robots' cameras and their motion through the environment. In our work, the second communication channel is established through a camera and a light.

There are multi-robot systems in which there is no explicit second communication channel such as [10, 6, 16], which still implicitly use one to solve the association problem. A common approach in such systems is to solve the association problem with the use of passive markers attached to the robots combined with a static precompiled list of marker-network address pairs. In these systems, the algorithmic aspect of the association problem is ignored and the system designer provides the static associations between robots and their IDs. See Fig. 1.2 for examples of such markers. These approaches, however, have limits. The deployment and redeployment of the markers is costly, the static list of associations needs to be updated and maintained by an external to the multi-robot system mechanism, which is typically the user. In practice, such marker systems are limited in the number of unique identifiers which can be encoded, suffer reliability issues, and have stricter restrictions on the maximum distance at which the markers can be identified. Such marker systems also impose significant computational and hardware requirements. Marker systems which scale with the system are especially undesirable as system size increases.

In our work we specifically address the algorithmic aspect of the association problem in a system in which the second channel is not powerful enough to allow for reliable and quick, direct transmission of IDs. Our algorithmic approach also has the advantage that it does not scale with the system size. Work done in [14, 15], which introduces the concept of *spatially targeted communication*, also approaches the algorithmic aspect of the association problem. Similarly to our work, the authors use the idea of a situated channel to target communication to a receiver based on their position in space. The model and solution described in the papers have similarities with our model and solution. However, the work in [14, 15] is focused on a narrower scope of the association problem. The authors have an approach which establishes an association between a pair of robots in the system, while we establish an association between all robots in the system in parallel, without restricting the use of the wireless and visual communication channels for any robot.

Mutual or collective localization in a multi-robot system, as exemplified by [5, 4] is related to the association problem because it solves a richer problem, the solution to which can be used to provide a common reference frame for robots in a system. However, the cost of localization of all robots may be greater than desirable just for targeted communication and these approaches do not specifically address the algorithmic aspect of the association problem.

2.2 Background

The **information space** and **information state** concepts introduced in [12] are at the basis of our algorithm analysis and understanding of the association process. An information state encapsulates the abstract information which a robot maintains to keep track of the physical world around it. The information space can be defined as the set of all information states and a transition function which describes how one information state is transformed into another, given an observation which encapsulates the information from sensing of the physical world.

We also use techniques based on discrete-time absorbing Markov chain [2] and discrete-time Branching Processes [9] to analyze and predict the macroscopic behavior of a multi-robot system solving the association problem using our algorithms. The Branching Processes method is used to predict the growth of populations of individuals. The individuals are divided into types and each has some known offspring distribution function. The properties of the individuals are used to predict how their population will grow. We apply this method to components of the internal state that a robot maintains about its neighborhood.

Some of the work presented in this thesis appeared in our paper, [11]. We expand the analysis presented in our previous work and offer an extended algorithm for solving the association problem for a system composed of mixture of stationary and mobile robots.

3. GENERAL DESCRIPTION OF THE ASSOCIATION PROBLEM*

3.1 Motivating Multi-Robot System

The motivating system consists of a large number of robots spread randomly and uniformly through an open and empty environment. The operation of the robots proceeds in discrete, synchronous time steps. Each robot in the system can directly observe and distinguish from the environment other robots using a camera or array of cameras with a 360° field of view. Robots are equipped with a light which can be on or off during any given time step. The camera and light form the situated visual communication channel. Robots can also broadcast messages through a wireless channel. Robots have limited communication range in both channels. Every robot is assigned a wireless ID and itself assigns visual IDs to robots it can detect with its camera. This motivating system will be used as the system on which we run simulation experiments and will serve as the basis of our formal model and analysis.

In this work, we do not consider the effects of communication noise or robot failure, as such, the communication channels are assumed to be perfect - no messages are lost or distorted. This assumption allows us to eliminate the impact of imperfect communication and to work on the pure association problem. The following is a list of assumptions about the motivating system.

Assumption 1. Perfect communication:

There are no lost, degraded or noisy messages. A message broadcast by a robot will be received by all robots within range during the same time step it was transmitted.

*Part of this chapter is reprinted with permission from P. Ivanov and D. A. Shell. Associating nearby robots to their voices. In *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALife)*, 2014, Copyright [2014] by Plamen Ivanov, Dylan Shell.

Assumption 2. Message size:

The wireless communication protocol is able to transmit the full binary representation of a robot's ID: $log_2(n)$ bits

The visual channel can transmit a message of size 1 bit

Assumption 3. Unlimited bandwidth and message processing:

Robots can receive and process any number of messages during each time step.

3.2 Formal Definition

Definition 1. A **multi-robot system** is a collection of robots which are equipped to communicate via one radio communication channel (channel 1) and one physically situated communication channel (channel 2). A multi-robot system can be described by the tuple:

1. V is a set of vertexes, each vertex represents a robot.

2. $G_1 = \langle V, E_1 \rangle$ and $G_2 = \langle V, E_2 \rangle$ are two directed graphs representing the connectivity of the system of robots over the two communication channels. $E_1 \subseteq V \times V$ has directed edges s.t. $e \equiv (v_i, v_j) \in E_1$. This is interpreted as robot i being able to receive a one-hop message from robot j over channel 1. $E_2 \subseteq V \times V$ is analogous, but for channel 2,

3. $E_2 \subseteq E_1$ — This is **Property 2**,

4. f_1, f_2 are labeling functions of the form $f : V \times V \mapsto C$ which are applied to channels 1 and 2, respectively. C is a set of labels.

5. C_1 and C_2 are the label sets for channels 1 and 2.

Each channel has an associated labeling function and the set of labels correspond to addresses (wireless and visual IDs, respectively). In the motivating system, the labeling functions will provide unique IDs for the wireless channel (IP addresses) and locally unique (defined precisely, next) IDs for the visual channel. The only requirement is that whatever procedure makes the ID assignment does not violate **Property 1**.

Property 1. Labeling function property:
$\forall x, y, z \in V, e_y = (y, x) \in E_{ch} \wedge e_z = (z, x) \in E_{ch} \Rightarrow f_{ch}(e_y) \neq f_{ch}(e_z)$.

The labeling function property states that for any 3 robots in the system, if a given robot can receive a message from any two robots through a given communication channel, then the ID of the two robots will not be the same. **Property 1** ensures that each neighborhood will have non-repeating IDs, which we call **locally unique IDs**. In the motivating system, it applies to both the wireless and visual IDs. Note that any labeling that provides globally unique IDs satisfies **Property 1**.

Property 2. Channel inclusion property:
When robot i can receive a channel 2 message from another robot then it is guaranteed to be able to receive a channel 1 message from that robot as well.

Property 2 ensures that a robot is guaranteed to receive a wireless message from robots in its visual range. This property is not required for solving the association problem, however, it simplifies the analysis by restricting the problem to a subset which is guaranteed to have a match for every visual ID in range. The algorithms and analysis can be extended to cover the full association problem space.

To examine an individual robot's perspective we introduce several additional concepts. We call the robot whose perspective we are examining the **observer** or

observing robot. Graphs representing the state of an individual robot and its neighborhood are depicted in Fig. 3.1.

Definition 2. A **neighborhood** is a subset of the system centered on an individual robot i. A neighborhood can be described by the tuple $\langle V^i, E_1^i, E_2^i, f_1,\ f_2, C_1^i, C_2^i \rangle$ where

1. i indicates the observing robot which is at the center of the neighborhood,
2. V^i consists of robots for which a directed edge $e \equiv (v_i, v_j) \in E_1$ exists,
3. E_1^i and E_2^i contain all directed edges in E_1 and E_2 which have robot i as source.
4. C_1^i, C_2^i are the labels given to edges in E_1^i and E_2^i by f_1 and f_2, respectively.

Definition 3. The **local view** of robot i is the information directly available to the robot. It can be described by the tuple $\langle n, C_1^i, C_2^i, O^i = \langle O_1^i, O_2^i, ...O_t^i \rangle \rangle$ where

1. n is the maximum possible robots in the entire system,
2. C_1^i and C_2^i are label sets as previously defined,
3. $O^i = \langle O_1^i, O_2^i, ...O_t^i \rangle$ is a sequence of observations (**Def. 5**) of the robots in the neighborhood of i, and t is the time step during which an observation was made.

Definition 4. An **activation** describes the communication activity of a neighborhood for a given time step t It is given by the tuple $A_t^i = \langle C_{1At}^i, C_{2At}^i \rangle$ where

1. C_{1At}^i is the set of labels of robots which are in range of and from which a message was sent to robot i on channel 1 during time step t,

2. C^i_{2At} is defined similarly for channel 2.

Definition 5. An **observation** is an individual robot's perception of an activation for a given time step t and is described by the tuple $O^i_t = \langle C^i_{1Ot}, C^i_{2Ot} \rangle$ where

1. C^i_{1Ot} is the set of labels of robots from which a message was received by robot i on channel 1 during the time step,

2. C^i_{2Ot} is defined similarly for channel 2.

Given the perfect communication assumptions, activations and their corresponding observations are identical.

Definition 6. Each robot maintains an **information state** [12], or **i-state**, which represents the current knowledge the robot has about its neighborhood. The i-state is described by the bipartite graph $S^i = \langle C^i_{1,1t}, C^i_{2,2t}, E_{1-2} \rangle$ where

1. C^i_{1t} and C^i_{2t} are subsets of the previously defined neighborhood label sets C^i_1 and C^i_2. The subsets consist of all labels from which a message was received in any time step up to t. These two subsets represent the two distinct vertex sets of the graph,

2. E_{1-2} is the edge set of the graph and represents possible associations between elements from each label set.

Each robot starts in complete ignorance of the actual one-to-one association of labels. As observations arrive, the observing robot updates its i-state by removing edges from the graph.

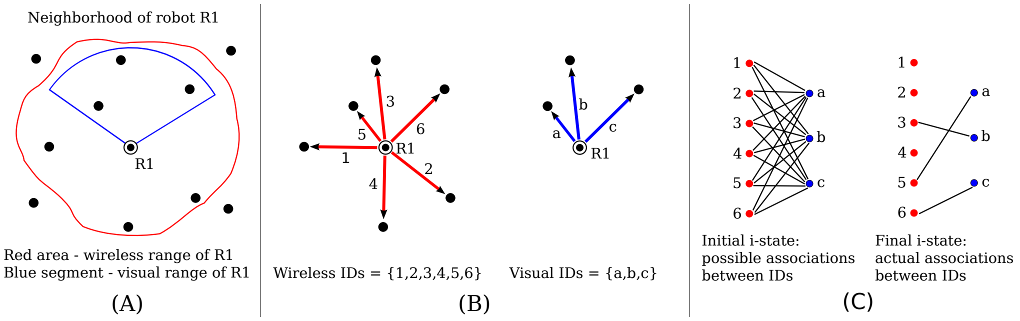


Figure 3.1: A robot and several neighboring robots as part of a multi-robot system are shown in (A). The wireless communication channel (in red) connects robot $R1$ to some of the robots surrounding it. The visual communication channel (in blue) has a more restricted view of robots in the system. In (B), we can see the neighborhood of robot $R1$ as given by Def. 2. The vertexes correspond to robots, the edges to the connectivity of $R1$ to its neighbors in each channel, and the label sets C_1^{R1}, C_2^{R1} are composed of the wireless and visual IDs. Two possible i-state bipartite graphs, as given by Def. 6 can be seen in (C).

Definition 7. The **communication channel association problem** for each robot i is:

Input: a local view with an observation sequence at time t,

Output: a complete or partial one-to-one association between channel 1 labels and channel 2 labels for the label tuple $\langle C_1^i, C_2^i \rangle$ corresponding to i's neighborhood at time t.

3.3 Information Space Interpretation

In this section, we expand upon our use of the information state and information space concepts, as described by [12].

A natural representation of the i-state of a robot is a bipartite graph. One set of vertexes in the graph is composed of the wireless IDs which are in range of an observing robot, the other set of vertexes is composed of the visual IDs which are in range. We denote the size of the wireless vertex set with w and the size of the visual vertex set with v. The edges of the graph represent possible associations between a wireless ID and a visual ID. The edges can be subdivided into two categories, fictitious edges and real edges. **Real edges** correspond to correct associations: they connect the wireless and visual IDs of the same robot. **Fictitious edges** on the other hand connect the IDs of two different robots and are therefore incorrect. A robot has no knowledge prior to eliminating an edge if it is fictitious or real.

An observation is also naturally represented by a bipartite graph. It contains the information the robot has collected about the possible associations between IDs during the current time step. The vertexes of an observation are equivalent to the vertexes of the i-state. Each edge represents an association between a wireless and a visual ID which have been always observed to have the same activity status. If an edge in the i-state graph has no corresponding edge in the observation graph, the edge has been identified as fictitious and is removed from the i-state. Real edges on the other hand will always remain part of an i-state graph, if both of their vertexes (IDs) are still part of the graph.

3.3.1 Structure of Individual I-States and Observations

An individual i-state can have any number of edges between v and $w \times v$. The initial i-state a robot starts with has the full number of edges as this represents no knowledge of the correct associations between wireless and visual IDs. The number of possible i-states, given w and v, is finite. This number is given by Bell's number [2] — it counts the number of ways a set of a given size, w, can be partitioned in. There are 2^w possible observation bipartite graphs, this corresponds to the number of ways w wireless IDs can be active or inactive.

An observing robot updates its i-state as new observations are made. Each observation partitions the vertexes of the i-state graph into two disjoint subsets — active and inactive IDs. The observations can be paired based on symmetry — the partitioning these observations produce is the same. All edges between vertexes which have different activity status are eliminated. Every subsequent observation can split each subset further. The final i-state is a disconnected bipartite graph which is composed of subgraphs of size 2 or 1. A subgraph of size 2 indicates an identified association between a robot's wireless and visual IDs. A subgraph of size 1 indicates a wireless ID whose robot is out of visual range. This process of elimination of fictitious edges allows an observer to solve the association problem.

Observations do not always result in elimination of edges. This can happen when an observation's partitioning has already been applied to the i-state. As the i-state gets sparser more and more observations bring no additional information. There are two observation which bring no information at any time: the observations corresponding to all or none of the robots being active. We call these the **zero-information observations**.

3.3.2 I-State Transition Graph

The i-states and observations for a given neighborhood form the **directed i-state transition graph**. The i-states form the vertex set of the transition graph and the directed edges correspond to the set of observations which lead from one i-state to another. A version of the transition graph can be seen in Fig. 3.2. The figure shows all i-states for a 4 × 3 neighborhood (4 wireless IDs and 3 visual IDs), all i-state to i-state transitions, and the number of observation for each transition.

The graph highlights some properties of the i-state space:

- The i-states can be grouped based on the number of partitions of the initial i-state bipartite graph they contain. We call these groups **levels**. The initial i-state is at level 1 (trivial partition of the graph S^i), and the final i-state is

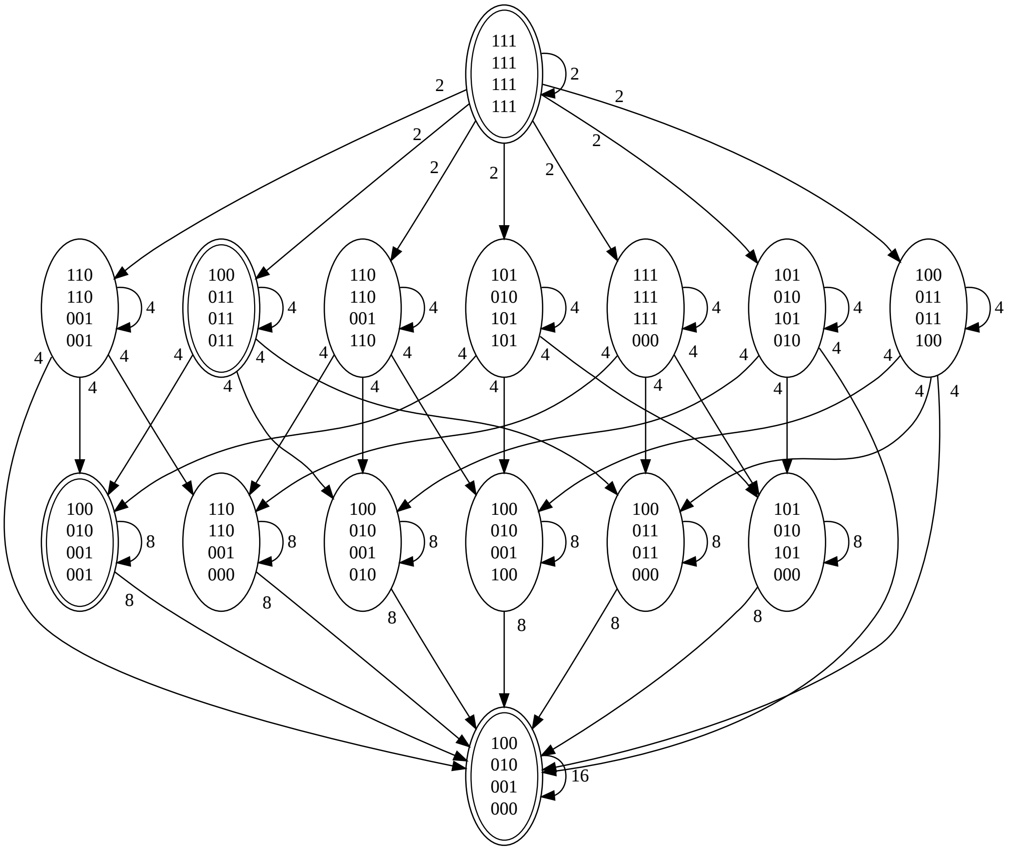


Figure 3.2: I-state transition graph and for the case of $w = 4$ wireless, $v = 3$ visual IDs. Each graph node contains an $w \times v$ i-state in matrix form and each edge is labeled with the number of observations which can trigger the given transition. The total number of observations for this case is 16. A specific path through the graph has been highlighted (double lined ovals) and a more detailed version can be seen in Fig. 4.1

at level w (there are w partitions of the graph S^i). All other i-states fall into levels 2 through $w - 1$.

- At each consecutive level the number of zero-information states doubles.
- Each i-state can transition to itself (through zero-information observations).
- There are no transitions between i-states from a higher partition level to a lower partition level and inside a level (with the exception of self-transitions).

Different algorithms allow a different subsets of observations which results in a different path through the information space. It is also important to note that the robots running any of the algorithms do not explicitly store, compute or use the information space which can be built for their current neighborhood. The information space idea is only used for understanding and predicting the algorithms' performance.

4. STATIONARY ROBOT CASE OF THE ASSOCIATION PROBLEM*

In this chapter we examine the stationary robot case of the communication channel association problem. The stationary case is concerned with a system of robots which are stationary (or very slow) and thus their relative positions do not change enough to affect their connectivity in either communication channel *i.e.,* there is no change in the set of robots an observer can receive messages from.

An example of this case is a multi-robot system of slow or stationary robots which need to establish the identities of their neighbors as part of an initialization phase. In some scenarios this can be achieved by using a static list of associations, however, in others, the spatial distribution of the robots is not known in advance. For such an example, take a collection of robots deployed from an aircraft or other moving vehicle with a deployment method which cannot position the robots in a pre-specified sequence.

To solve the stationary case we introduce 3 algorithms. The first two are deterministic and are mainly used as a point of comparison for the third algorithm, which is probabilistic in nature, hence its name.

4.1 Algorithms for Solving the Stationary Association Problem

For the stationary case, we make the following assumption: every robot knows the number of robots in its neighborhood. In general systems this assumption can be made to hold by adding an initial time step in which all robots transmit a message on both channels. The assumption is not required for the operation of the presented

*Part of this chapter is reprinted with permission from P. Ivanov and D. A. Shell. Associating nearby robots to their voices. In *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALife)*, 2014, Copyright [2014] by Plamen Ivanov, Dylan Shell.

algorithms, but allows us to analyze the performance more uniformly by establishing the same starting i-state for all cases: the fully unresolved one.

4.1.1 Naïve Algorithm

In the Naïve algorithm, every robot emits a visual message during the time step which is equivalent to the robot's wireless ID. An observer simply fills a table associating a wireless ID to a visual ID which is currently active. The pseudo code for this algorithm can be seen in **Algorithm 1**.

Algorithm 1 Naïve Algorithm

```
procedure NAÏVE
    matchTable                                   ▷ [wireless ID, visual ID] pairs
    for $t \leftarrow 1, n+1$ do                 ▷ For each possible wireless ID
        if $t - 1 == myID$ then
            $broadcast(visualMessage)$
        end if
        if $messagesReceived()$ then
            if $local\_robot.isActive()$ then
                $pair \leftarrow [t-1, visualID]$
                $matchTable.add(pair)$
            end if
        end if
    end for
end procedure
```

4.1.2 Logarithmic Algorithm

In the Logarithmic algorithm the fact that n identifiers can be represented with $\log_2(n)$ bits is used. The observer associates each observable robot with an array of size $\log_2(n)$. At time step t, all robots whose t^{th} bit is equal to 1 will emit a visual message. After each time step an observer can fill one more bit in the array of bits corresponding to the wireless IDs of the visually active robots. The pseudo code for this algorithm can be seen in **Algorithm 2**.

Algorithm 2 Logarithmic Algorithm

```
procedure LOGARITHMIC
    self_bit_arr ← to_bit_array(my_ID)
    table_arr ← array[num_neighbors][$\log_2(n)$]
    for $t \leftarrow 1, \log_2(n) + 1$ do
        if self_bit_arr[$t - 1$] == 1 then
            broadcast(visualMessage)
        end if
        for $j \leftarrow 0$, num_neighbors do
            if neighbor[$j$].isActive() then
                table_arr[$t$][$j$] = 1
            end if
        end for
    end for
end procedure
```

4.1.3 Probabilistic Algorithm

In the Probabilistic algorithm all robots construct a $w \times v$ i-state matrix to represent their i-state. Here w represents the number of robots in direct communication over channel 1 (C^i_{1t} from S^i) and v — the number of robots over channel 2 (C^i_{2t} from S^i). The i-state starts filled with 1's, which indicates complete ignorance of the one-to-one associations. Each cell in the i-state matrix corresponds to an edge in E_{1-2} from S^i. The wireless IDs (labels) of all robots that are within communication range are associated to an activity array (C^i_{1Ot} from O^i_t) that keeps track of which robot transmitted a message in the current time step (1 is used to denote a transmission). Each wireless ID is given an index $h \in [0, w-1]$. Similarly, the visual IDs are associated with an activity array (C^i_{2Ot} from O^i_t). Each visual ID is given an index $s \in [0, v-1]$. At any time step the i-state matrix contains the observer's current information about the matching between wireless and visual IDs.

During each time step, each robot emits a **message pair** (simultaneous transmission in both channels) with 50% probability. Every observer fills in a $w \times v$ observation matrix, which captures an observation of the neighborhood made during the current time step. This matrix is filled based on the following principle: the $[h][s]$ entry in the observation matrix is filled with a 1 if both the h^{th} wireless ID and s^{th} visual ID were active or inactive during the current time step. All other entries are filled with 0's. Once the observation matrix has been constructed it is used to update the i-state matrix of the observer. All entries in the i-state matrix are multiplied by the corresponding entry in the observation matrix. This produces the updated i-state matrix. When the i-state matrix contains only v 1's, the association problem for the observing robot is complete. The non-zero $[h][s]$ entries in the matrix tell us that the h^{th} wireless ID and the s^{th} visual ID belong to the same robot.

The performance of the probabilistic algorithm scales with the size of robots' neighborhoods and not with the size of the full system. Every robot can quickly and in parallel establish the communication channel association for its neighborhood. Pseudo code for the Probabilistic algorithm appears in **Algorithm 3**, and an i-states and observations for a single run of the algorithm for one neighborhood are depicted in Fig. 4.1.

4.2 Analyzing and Predicting the Performance of the Probabilistic Algorithm

Our focus is mainly on analyzing and predicting the performance of the probabilistic algorithm as it is the one which scales locally. We will not treat the two deterministic algorithms with the full analysis.

Algorithm 3 Probabilistic Algorithm

```
procedure PROBABILISTIC
    broadcast(messagePair)
    $w$                                  ▷ number of wireless IDs in range
    $v$                                  ▷ number of visual IDs in range
    $istate[w][v] \leftarrow 1's$
    $obs[w][v]$
    $w\_activity[w]$                     ▷ 0 - inactive, 1 - active
    $v\_activity[v]$                     ▷ 0 - inactive, 1 - active
    for each time step do
        if $fairCoinFlip()$ then
            $broadcast(messagePair)$
        end if
        $w\_activity \leftarrow receivedWifiMsgs()$
        $v\_activity \leftarrow receivedVisMsgs()$
        $obs \leftarrow fillObs(w\_activity, v\_activity)$
        $istate \leftarrow updateState(obs)$
        if $istate.isFinal()$ then
            $ready \leftarrow TRUE$
        end if
    end for
end procedure
```

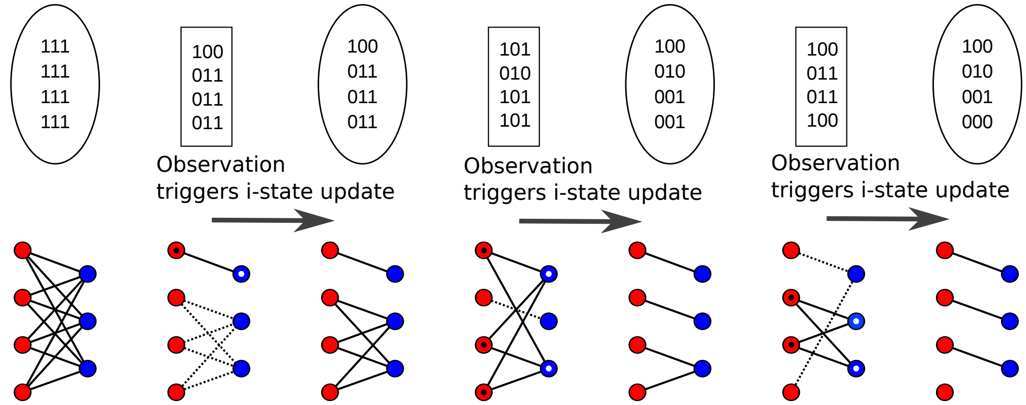


Figure 4.1: An i-state path through the transition graph. Here we can see the i-state of the robot (the matrices in ovals) and its progression as the robot makes observations (the matrices in squares). Bipartite graphs corresponding to each i-state and observation appear at the bottom. The set of 4 vertexes represents the wireless IDs and the set of 3, the visual ones. In an observation graph, a vertex with a black/white mark in it denotes an ID which is transmitting a message during the current time step. The solid edges in the observation graph connect currently active IDs, while the dashed edges connect inactive IDs. This path is highlighted in the full i-space graph in Fig. 3.2.

4.2.1 Performance Metrics

We define two performance metrics for quantifying how successful the probabilistic algorithm is at completing the stationary communication channel association problem. The first metric, **expected time to completion**, measures the number of time steps the algorithm took to complete the association problem for a given neighborhood. The second metric, **fraction matched**, measures the fraction of visial IDs which have been unambiguously associated (matched) with a wireless ID.

4.2.2 Expected Time to Completion

4.2.2.1 Information Space and Absorbing Markov Chain

The Probabilistic algorithm's path through the information space graph is not determined by the specific IDs in a given neighborhood and it can include any valid transition in the i-space. Since robots running the Probabilistic algorithm transmit a message pair with 50% probability, all possible observations are equally likely. Given the likelihoods of different observations, we can construct a transition probability matrix that corresponds to the information space. You can see the transition matrix for the information space from Fig. 3.2 in Fig. 4.2. The information space graph has the necessary properties to be considered an absorbing Markov chain [7] — it has an absorbing state (the final state) and this state can be reached from any other state in the graph (transient states). The absorbing Markov chain allows us to compute the expected number of steps from any state in the graph to the absorbing state given the transition matrix — this would be equivalent to the expected time to completion for the probabilistic algorithm. To compute this we use the following formula:

$$\mathbf{t} = (I - Q)^{-1}\mathbf{1} \tag{4.1}$$

Here Q is a $t \times t$ matrix containing all transient to transient transition probabilities, $\mathbf{t}$ denotes a vector of size t, I is the identity matrix, and $\mathbf{1}$ is a column vector of all 1's. The i^{th} element of $\mathbf{t}$ gives the average expected number of steps to absorption if the process started from the i^{th} transient state. Thus, Eq. (4.1) can be used to calculate the expected running time to complete the association problem for a given neighborhood.

This approach, however, is prohibitively expensive for all but the smallest values of w. As was noted earlier, the number of i-states for a given neighborhood is given by Bell's number: B_w. The authors of [2] give an upper bound of Bell's number which is $O(w!)$ and $\Omega(2^w)$ for a set of size w. A simple application of all possible observations to all possible i-states can naïvely generate the information space transition probability matrix in $O(2^w B_w)$ time. The matrix inversion needed

1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8	0	0	0	0	0	0	0
0	1/4	0	0	0	0	0	0	1/4	1/4	1/4	0	0	0	0
0	0	1/4	0	0	0	0	0	1/4	0	0	1/4	1/4	0	0
0	0	0	1/4	0	0	0	0	1/4	0	0	0	0	1/4	1/4
0	0	0	0	1/4	0	0	0	0	1/4	0	1/4	0	1/4	0
0	0	0	0	0	1/4	0	0	0	1/4	0	0	1/4	0	1/4
0	0	0	0	0	0	1/4	0	0	0	1/4	1/4	0	0	1/4
0	0	0	0	0	0	0	1/4	0	0	1/4	0	1/4	1/4	0
0	0	0	0	0	0	0	0	1/2	0	0	0	0	0	1/2
0	0	0	0	0	0	0	0	0	1/2	0	0	0	0	1/2
0	0	0	0	0	0	0	0	0	0	1/2	0	0	0	1/2
0	0	0	0	0	0	0	0	0	0	0	1/2	0	0	1/2
0	0	0	0	0	0	0	0	0	0	0	0	1/2	0	1/2
0	0	0	0	0	0	0	0	0	0	0	0	0	1/2	1/2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 4.2: This figure shows a transition probability matrix constructed from the information space transition graph of a neighborhood with $w = 4$ and $v = 3$.

in Eq. (4.1) will take $O({B_w}^3)$. The absorbing Markov chain method, therefore, runs in $O(w!^3)$. Clearly, a cheaper approach predicting the time to completion is needed.

The advantage of the absorbing Markov chain method is that it gives very accurate predictions — in a comparison between several expected times we computed with the method and simulation results there was no difference in the values up to the precision we used for the calculation.

4.2.2.2 Simulation Results

To get an idea of the performance of the Probabilistic algorithm in terms of the time to completion metric, we simulated a number of neighborhoods and gathered statistics on the time it took the algorithm to complete the association problem.

The set up for the simulation was as follows. A w and v were chosen, after which an activity array was created for the wireless IDs and a full i-state matrix created. The w and v values correspond to the number of robots in wireless and visual communication range for a real robot. The visual IDs were matched from first to last to the first v available wireless IDs, thus making the real edges occupy the diagonal of the i-state matrix. Next, each neighborhood was run through the Probabilistic algorithm with random activations being generated for the wireless and visual ID activity. Each neighborhood configuration was run until completion or until a threshold was reached. The experiment was repeated 50,000 times for every w and v combination that was examined. We simulated all neighborhood size combinations of $w \in [2, 85]$ and $v \in [1, w-1]$.

To validate the data from the simulation we also calculated the precise expected value for all combinations of $w \in [2, 8]$ and $v \in [1, w-1]$ using the absorbing Markov chain method, results from which matched the simulation up to the precision we used during the calculations (0.0001 time steps). The results from the simulation trials of neighborhoods up to size 40 can be seen in Fig. 4.3, the remaining trial data is omitted to keep the figure more readable. A more detailed breakdown of the performance of the algorithm can be seen in Fig. 4.4. In the figure we have plotted the distribution of times to completion for four w and v combinations.

From the distribution of results we can see that most solutions are concentrated around the mean value. We can also see that there is a very long but thin tail to the distribution, as can be expected for a probabilistic process which can take any number of steps to complete. From the data we can also see that there is a minimum time to completion as well. This minimum time is equal to $log_2(v)$. This is the smallest number of steps we need to take to split a set into 1-element subsets if we can split every subset at each time step. The time to finish splitting is equivalent to the depth of a balanced binary tree which has v leaf nodes.

4.2.2.3 Branching Processes Model

Branching Processes [9] are used to model populations of individuals which go through reproductive cycles at given intervals (for discrete time branching processes). Individuals from the population can reproduce and create some number of offspring in the next reproductive cycle. The number of offspring is described by some known distribution. The individuals themselves can be modeled as living only one repro-

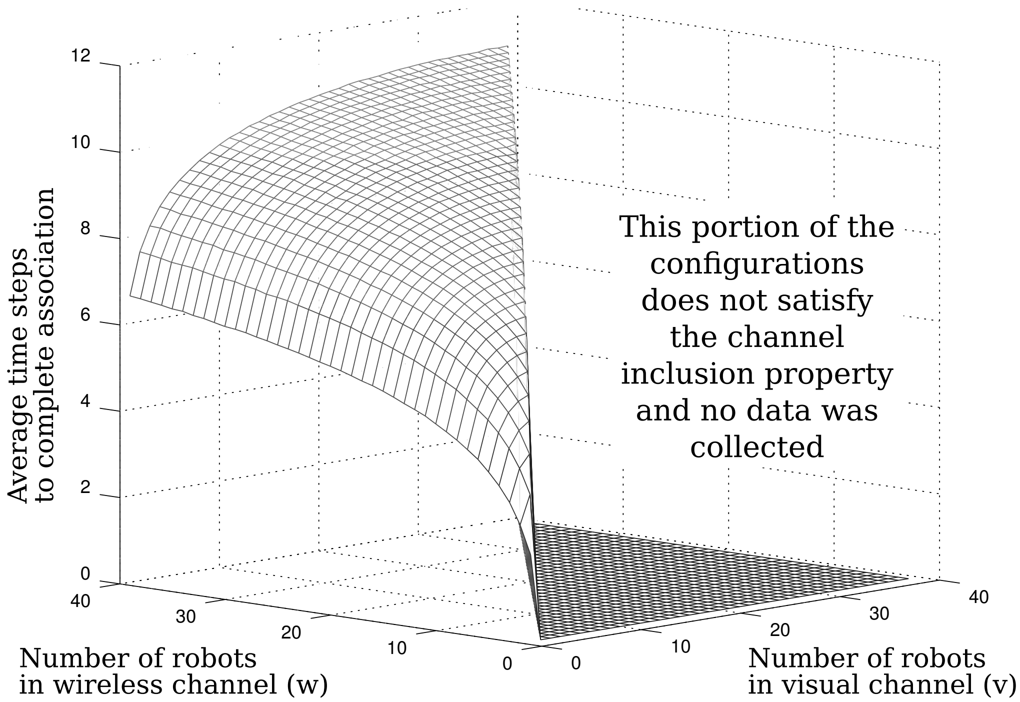


Figure 4.3: This is a plot of the average number of time steps to complete the association problem for all $w \times v$ neighborhoods for which $w \in [2, 40]$ and $v \in [1, w]$. Every neighborhood was simulated 50,000 times to obtain the results. The area with no data corresponds to neighborhoods in which $w < v$, which violates the channel inclusion property, so no data was collected.

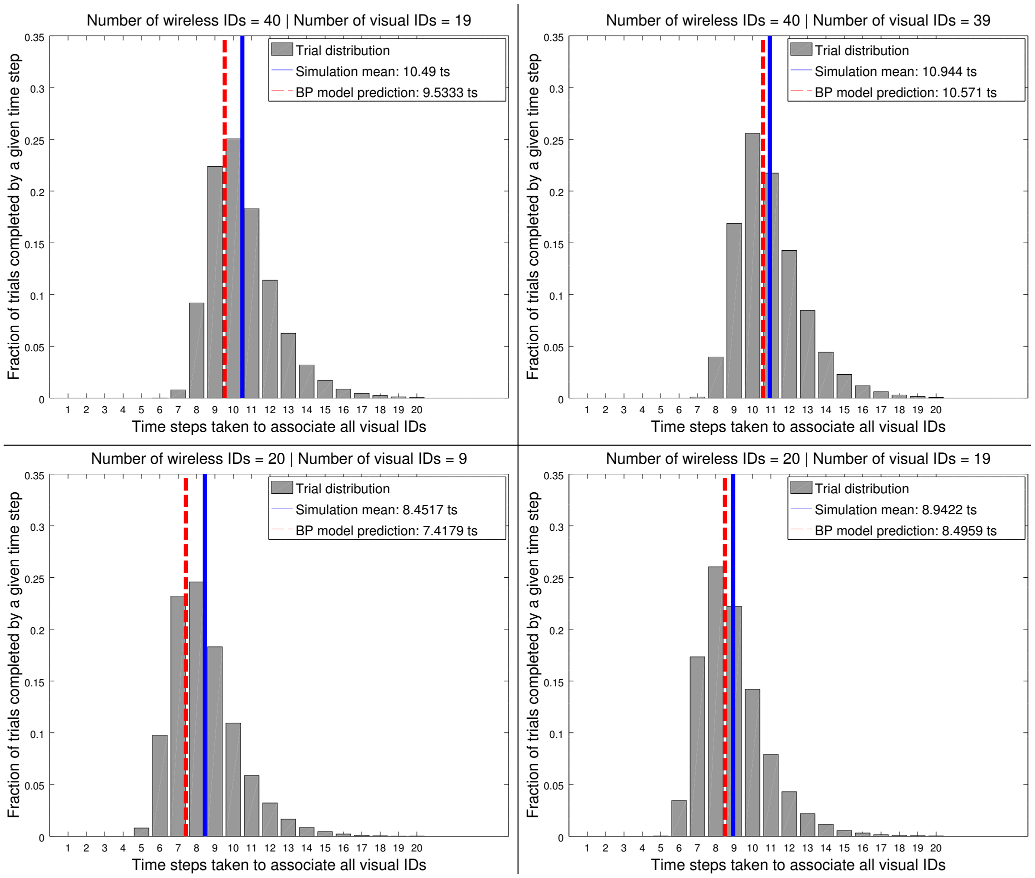


Figure 4.4: This figure shows a plot of the distribution of the different times to completion of the association problem for 4 combinations of w and v. It also includes the mean number of time steps as calculated from the simulation data (blue vertical line) and the Branching Processes model prediction (red vertical line).

ductive cycle. Using the Branching Processes model, we can answer questions such as:

- "Will this population go extinct?"
- "How long will this population take to go extinct?"
- "What is the expected number of individuals in a given reproduction cycle after the start of the population?"

Given the above description, we can define as a population the total number of potential associations in the i-state of a given robot. Each wireless to visual ID pairing can be modeled as an individual, which can either produce 1 offspring (itself, as it has not been eliminated by observation) or 0 offspring (be discarded as it has been eliminated by an observation). Once a population reaches some number of individuals (the number of visual IDs which the current robot can observe) we know that the association problem for the given robot's neighborhood has been solved.

Not all potential associations have the same likelihood of producing offspring. For example, real edges, unlike the ones we described above, will always produce 1 offspring. This can add inaccuracies to the calculated time to extinction. The BP model can be extended to handle this situation by introducing the idea of multiple types in a population. Each type has a different distribution of offspring associated to it. This allows more accurate representation of the association process.

Keeping track of the mean number of individuals as time passes can be done simply by using the following relationship: Nm^t where N is the initial population, m is the mean number of offspring and t is the current generation since starting the process. For the fictitious edges, $N = wv - v$ and $m = 0.5$. Here the value of m is determined by the probability of two unrelated IDs to have different activity status at any given time step, which is 50 %, hence the mean number of offspring for a fictitious edge is 0.5. For real edges, $N = v$ and $m = 1$. This allows us to keep track of the expected number of fictitious edges at every time step. The Branching Processes literature offers equations for calculating the approximate expected time to extinction, *i.e.,* the number of generations until the population has completely disappeared. An applicable approximation, from [9], for the expected time to completion is given by Eq. (4.2).

$$E_N[\tau] = \frac{\ln(N)}{|\ln(m)|} = \frac{\ln(wv - v)}{|\ln(0.5)|} \tag{4.2}$$

Fig. 4.5 and Fig. 4.6 show a comparison between simulation data and the Branching Processes model approximation for two numbers of wireless IDs and several possible visual ID counts. As can be seen the approximation underestimates the actual values, but falls well within the bulk of the solution lengths. It can also be noted that the approximation is closer to the simulated value for higher v values. Despite some inaccuracy, the Branching Processes model is a cheap method for predicting the expected time to completion for any neighborhood in the stationary association problem. In addition, the prediction is based on a model of the association process and therefore reflects it more accurately than simple data fitting, for example.

4.2.3 Fraction Matched

4.2.3.1 Simulation Results

The results for the second metric, fraction matched, can be seen in Fig. 4.7. The figure contains an average over several simulation runs for neighborhoods of different sizes. The simulation set up was the same as the one in the previous section. In addition, to the time to completion, we kept track of the number of visual IDs unambiguously associated with a wireless ID. From this we computed the expected

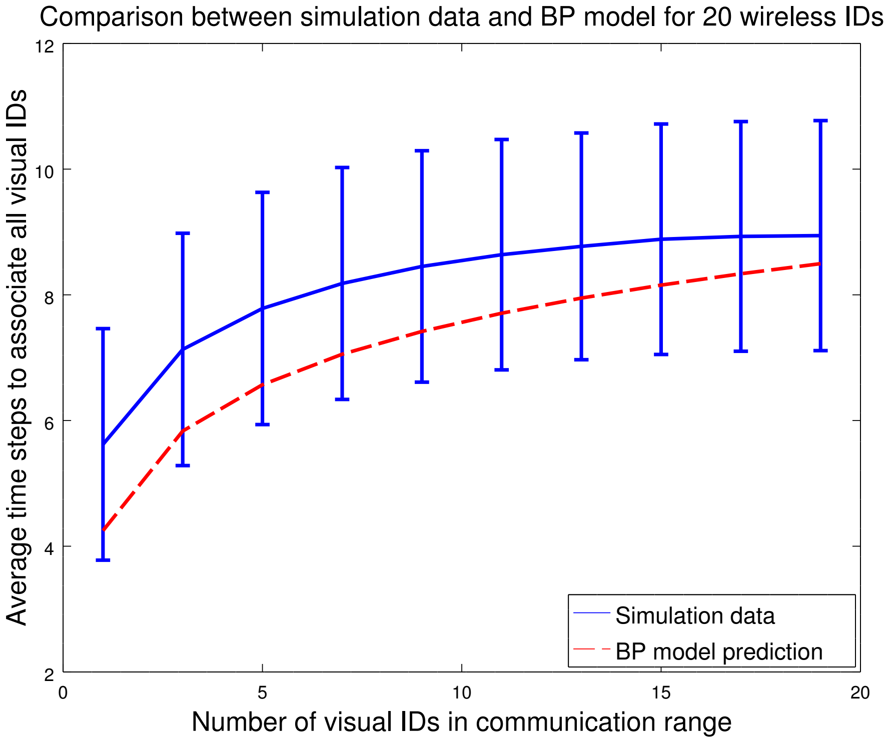


Figure 4.5: This figure shows the mean and 1 standard deviation above and below the mean of the expected time steps to completion of the association problem for $w = 20$ and $v \in [1, w-1]$ in increments of 2. The prediction calculated with the Branching Processes model is given in dashed red.

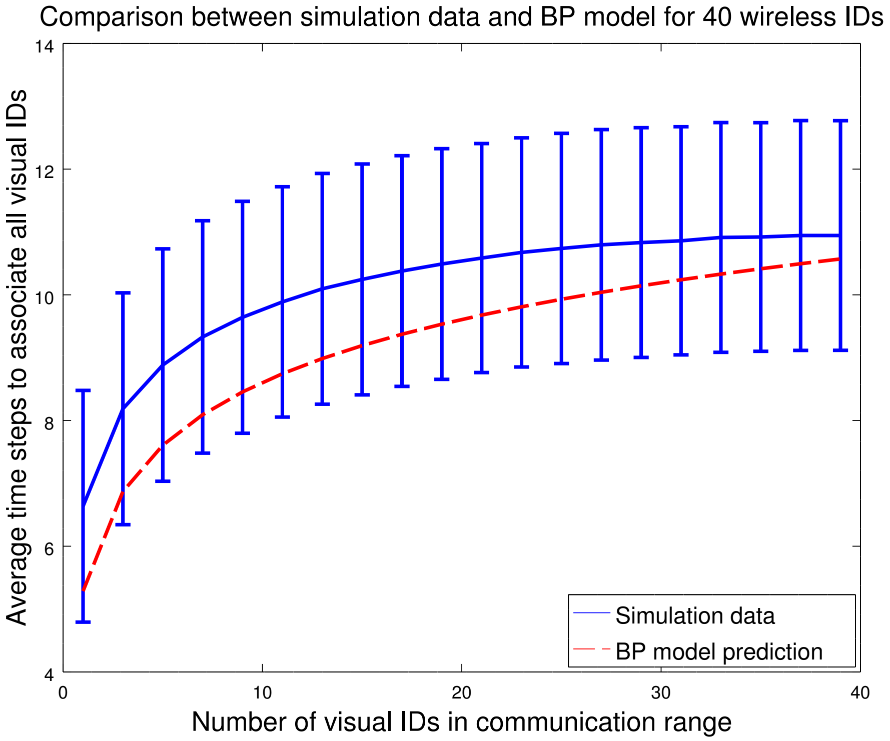


Figure 4.6: This figure shows the mean and 1 standard deviation above and below the mean of the expected time steps to completion of the association problem for $w = 40$ and $v \in [1, w - 1]$ in increments of 2. The prediction calculated with the Branching Processes model is given in dashed red.

fraction of visual IDs which are solved at each time step.

An important observation made from the collected data on the fraction matched metric is that the fraction of unambiguously associated visual IDs as time passes depends only on the number of wireless IDs in the neighborhood. This means that the expected behavior of the fraction solved as time steps progress when the channel inclusion property holds is equivalent for all $v \in [1, w]$. This allowed us to create a simple predictive model for the fraction solved metric which takes as input w and t (time steps since start of algorithm) and outputs the expected fraction matched for the given number of wireless IDs and time steps since starting the algorithm.

4.2.3.2 Fraction Matched Model

The fraction matched model works as follows. We assume that $w = v$ and calculate the starting number of fictitious edges as $wv - v$. Then we use the relationship for expected number of fictitious edges with time as given in Eq.(4.3) to produce a sequence of expected fictitious edges for each time step.

$$F_t = F_{\text{init}} \times m^t \tag{4.3}$$

Next, for every visual ID we compute the probability it will have zero fictitious edges assigned to it. This represents a visual ID which has been unambiguously associated. The probability that a visual ID has zero incoming fictitious edges is given by the probability that all fictitious edges have been assigned to any of the rest of the visual IDs. This probability is given by Eq. (4.4). The equation also gives us the fraction of the visual IDs which have been unambiguously associated.

$$P_{matched} = \left(\frac{v-1}{v}\right)^{F_t} \tag{4.4}$$

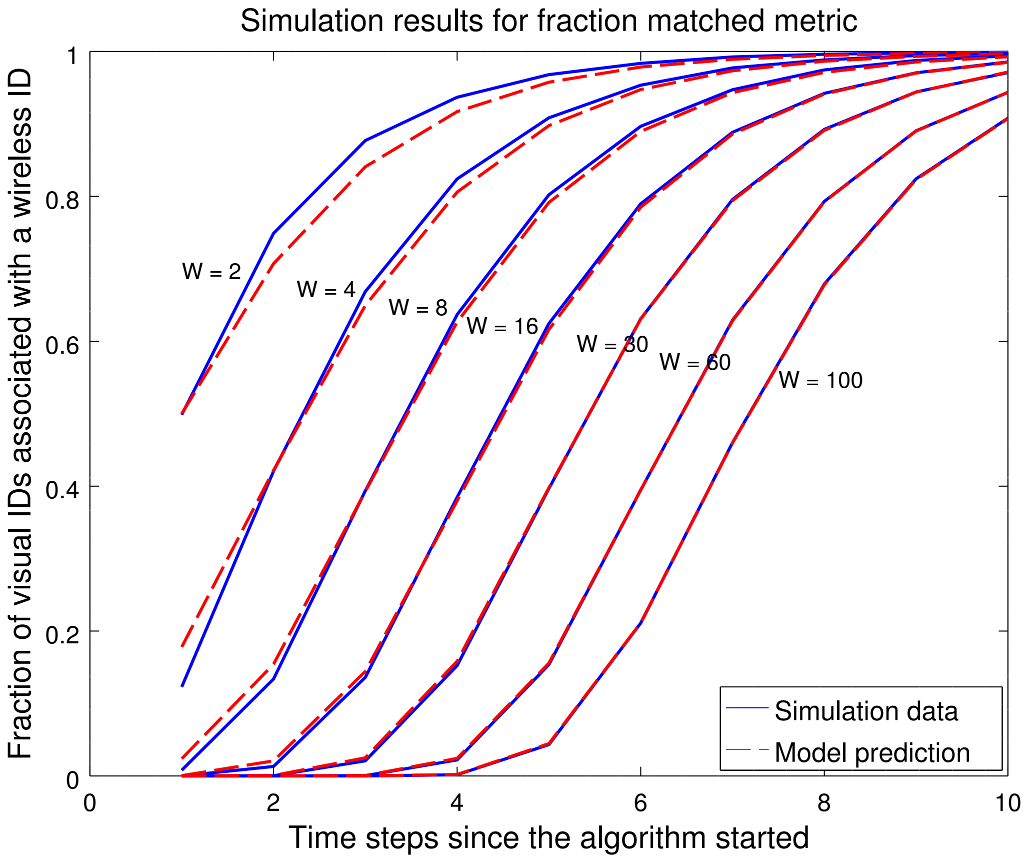


Figure 4.7: This figure displays the fraction matched metric from simulation data of different number of wireless IDs (w) as well as the predictions of our fraction matched model.

Since the process of associating visual and wireless IDs is probabilistic and combinatorial in nature, we made several assumptions. First, we assumed that each visual ID is independent from the other visual IDs. Second, we assumed that there is no limit to how many fictitious edges can be assigned to the other visual IDs, when looking from the perspective of a single visual ID. A comparison between the fraction matched model prediction and the simulated data can be seen in Fig. 4.7.

4.2.4 Comparing the Algorithms

We have focused primarily on the Probabilistic algorithm because it is harder to predict and, unlike the two deterministic algorithms, scales locally. In this situation local scaling means that the time to solve the association problem grows with the size of a robot's neighborhood and not with the system size.In this section we show how the three algorithms compare in time to completion and number of messages sent. This shows that despite the better scaling of the Probabilistic algorithm, the deterministic ones can be more advantageous if the system properties are appropriate.

The Naïve algorithm runs for n time steps, where n is the number of unique IDs that can be used by the system of robots. The algorithm runs for n time steps regardless of how many of the n identifiers are in use by actual robots. Every agent emits only one message during the execution of this algorithm — at most n for the entire system. The Logarithmic algorithm runs for $\log_2(n)$ time steps and generates at most $\frac{1}{2}n\log_2(n)$ messages. This is due to the fact that at most half of all robots will emit a message pair at each time step, since only half of all possible unique IDs will have a 1 in their i^{th} bit. The Probabilistic algorithm runs on average for $\frac{\ln wv-v}{|\ln 0.5|}$ time steps and generates an expected $0.5n\frac{\ln wv-v}{|\ln 0.5|}$ messages, if we assume that the robots stop transmitting once all their neighbors have solved the association problem.

The two deterministic algorithms have a performance which is dependent on the global size of the system. They also provide an exact upper bound on execution time for the association problem. The Naïve algorithm gives us the slowest performance but uses the least amount of messages. The Logarithmic algorithm is faster than the Naïve one but requires an increase in the number of messages used. The two deterministic algorithms also require minimal use of the wireless channel (only used during the first system wide transmission, which is not strictly required for the

Logarithmic algorithm). The probabilistic algorithm does not depend on the global size of the system. Every robot can quickly achieve the communication channel association for its neighborhood. This algorithm has the disadvantage that it has no guaranteed time expectancy. The overall performance of the system is dependent directly on the types of neighborhoods that compose it so an analysis of the robot distribution may be required before choosing this algorithm.

5. MOBILE ROBOT CASE OF THE ASSOCIATION PROBLEM

In this chapter we focus our attention on the mobile case of the communication channel association problem. In this scenario, the robots comprising the system are moving fast enough for the connectivity between them to change potentially during every time step of the algorithm's execution. This creates new problems for any potential algorithm which is solving the problem and requires changes to the analysis of the probabilistic algorithm's macroscopic behavior. The mobile case is what most realistic robot system would look like while they operate.

5.1 Information State and Information Space

The idea of the information state as part of an information space is still valid for the mobile version of the Probabilistic algorithm, however the information state and information space sizes and structures can now change during every time step of the algorithm. Since IDs can both be added and removed, a new subprocess is added to the communication channel association problem. It is no longer practical to use the two concepts in predicting the behavior of a system of robots running the algorithm, hence we no longer use the information space explicitly in our analysis.

Looking at the information space landscape, we can see that the algorithm's operation is trying to move from i-states of less knowledge to i-states with more knowledge (more unambiguously associated visual IDs). On the other hand, the random process of adding and removing IDs traverses the i-space in both directions: decreasing or increasing knowledge. On average, the traversal due to the random process results in decreasing the knowledge of the observing robot.

The mobile case of the association problem can then be imagined as a two part process. One subprocess acts multiplicatively to reduce the number of fictitious edges. This subprocess is the elimination of edges in result to conflicting observations. The second subprocess acts additively to increase the number of edges by introducing new edges to the process. Eq. 5.1 models the structure of the two part nature of the association problem.

$$T_t = (T_{t-1} \times m) + N_t \tag{5.1}$$

In Eq. 5.1, T_t denotes the number of total edges for time step t, T_{t-1} denotes the number of edges in the previous step, and N_t denotes the number of newly added edges and m denotes a multiplicative factor, $m < 1$, for the dependence of edges from one time step to the next.

Eq. (5.1) indicates a relationship between the mobile and stationary cases of the association problem. The stationary case only has the multiplicative subprocess, while the mobile case incorporates the stationary dynamics and adds the additive subprocess. As such the stationary case of the association problem is a subtype of the mobile case of the association problem.

5.2 Algorithm for Solving the Mobile Robot Case

The probabilistic algorithm from chapter 4 requires some modifications in order to handle the changing nature of the robot's neighborhood. We need to introduce the idea of an evolving information state. We need to distinguish between new and old IDs and we need to remove IDs which are no longer in range. Unlike the stationary case, we no longer assume that the full number of robots (IDs) which are in range is known in advance. The assumption of knowing the full numbers does not provide a significant benefit to the algorithm and its analysis.

A new wireless ID enters an observer's range as soon as a message from it is received. This can only happen when the ID is in range and active. Visual IDs on the other hand can enter the range of an observer before they activate since the robot can distinguish a neighbor even when it is not transmitting through the visual channel. However, since adding a visual ID before it is activated brings no practical information about the potential candidate wireless IDs for it, we ignore visual IDs which are inactive and only add them to the observer's i-state once they activate (transmit a message in both channels). This, given our earlier assumptions, guarantees that there will be at least one candidate wireless ID to be associated with the visual ID.

Visual IDs can simply be removed when they leave the robot's range. This is not applicable to the wireless IDs since an observing robot cannot detect when a robot has left its wireless range. Thus the observer needs to keep track of how long a wireless ID has been inactive. After a preset number of time steps if an ID has been inactive, it is dropped from the list. By choosing an appropriate drop time, the robot limits the number of IDs it needs to keep track of and also guards against losing too much information by dropping IDs too soon.

The algorithm can operate without dropping wireless IDs, however, this runs the risk of creating unnecessary work with little or no benefits. For example, if a robot is likely to see a large fraction of all IDs in the system, but will only be in communication range with relatively few of them, then there is no need to keep old wireless IDs around. Since the algorithm operates by adding potential edges between active IDs only, a wireless ID which has been encountered before and dropped will be added back as soon as it is detected. Furthermore, visual IDs are dynamically assigned to the robots in range by the observing robot. This means, that a wireless ID which re-enters communication range is unlikely to be matched with the same visual ID, making keeping the old wireless ID around even less useful.

Another major change to the Probabilistic algorithm is a procedure which modifies the i-state based on the list of new and removed IDs before it updates the i-state using the current time step's observation. The procedure, removes rows and columns from the i-state corresponding to wireless and visual IDs which have exited their respective ranges. The procedure also adds new rows and columns into the i-state for new wireless and visual IDs. New IDs increase the size of the vertex sets of the i-state. Once new IDs have been added, the appropriate edges between vertexes need to be added to the i-state as well. Edges are added as follows. An edge is added between all new wireless and new visual IDs. In the case of equal wireless and visual range, no more edges are added. In the case of the wireless range being greater than the visual range, new edges are added between all new visual IDs and old, not yet associated wireless IDs. This is necessary because a robot can enter wireless communication range much sooner than it will enter the visual communication range of an observer. This means that a newly added visual ID may be matched with a wireless ID which has already been included in the i-state of the observing robot. In this work, we only examine the case where the wireless range is equal to the visual range.

The pseudo-code for the mobile version of the Probabilistic algorithm can be seen in **Algorithm 4**.

5.3 Performance Metrics

Given that there is a positive additive component to the association process of the mobile association problem, we can no longer use the *expected time to completion* metric as the i-state is unlikely to achieve a fully associated status. Instead we introduce a new metric — the **ratio of total to real edges**. This metric gives us a measure of how ambiguous the current i-state is by counting the average number of real and fictitious edges for every visual ID. Clearly a ratio of 1 means we have no ambiguity and a higher ratio means, we have more ambiguity. The *fraction matched*

Algorithm 4 Modified Probabilistic Algorithm

```
procedure PROBABILISTIC MOBILE
    $w; v$    ▷ number of Wireless/Visual IDs in list
    $w\_ids; v\_ids$    ▷ current list of W / V IDs
    $w\_new; v\_new$    ▷ newly added W / V IDs
    $w\_solved; v\_solved$    ▷ Solvedness Status of W / V IDs
    $w\_activity; v\_activity$    ▷ Activity status of the W / V IDs
    $w\_inactivity;$    ▷ Time since last activation
    $drop\_time$    ▷ Time steps before a W ID is dropped
    $w\_current; v\_current$    ▷ All W / V IDs which are currently active
    $v\_lost$    ▷ V IDs which exited the visual range
    $istate[h][s] \leftarrow 1's$
    $observ[h][s]$
    for each time step do
        if $fairCoinFlip()$ then
            $broadcast(messagePair)$
        end if
        $w\_current \leftarrow receivedWifiMsgs()$
        $v\_current \leftarrow receivedVisMsgs()$
        $v\_lost \leftarrow exitedRangeVisIDs()$
        $removeVisIDs(v\_lost)$    ▷ Remove the V IDs from data structures
                                   ▷ Mark any matched W ID as unsolved
        $removeWifIDs(drop\_time)$    ▷ Remove W IDs which reach drop time limit
                                   ▷ mark any matched V ID as unsolved
        $updateExisitingVisIDs(v\_current)$    ▷ Update activity of existing V IDs
        $addNewVisIDs((v\_current)$    ▷ Add the new V IDs to the structures
        $modifyIStateVis(v\_new)$    ▷ Expand the istate
        $updateExisitingWifIDs(w\_current)$    ▷ Update activity of existing W IDs
        $addNewWifIDs(w\_current)$    ▷ Add the new W IDs to data structures
        $modifyIStateWif(w\_new)$    ▷ Expand the istate matrix
                                   ▷ Add 1's (edges) between new W IDs and new V IDs
        $modifyObservation$    ▷ Expands the observation matrix to match the istate
        $observ \leftarrow fillObs(wifi\_activity, vis\_activity)$    ▷ update Observation matrix
        $istate \leftarrow updateIState(istate, observ)$
                ▷ Multiply the istate and observation (by element) to create current istate
    end for
end procedure
```

metric is still applicable.

5.4 Experimental Set Up

The multi-robot system we simulate is set up as a mixture of two types of robots: stationary and mobile. The majority of the robots (less than 10% are mobile) in the system are stationary. They are placed on a grid in the environment with regular separation. The stationary robots are also uniformly perturbed from their grid position within half a grid spacing in either direction along each axis. This creates a random distribution of robots but maintains a uniform density of robots in the environment. The mobile robots are positioned uniformly and randomly in the entire environment and are given a random direction in which they move. All mobile robots share the same speed. The mobile robots continue moving for the duration of the simulation. When they encounter the environment boundary, they change direction by bouncing off the boundary. The mobile robots are used to collect data about the performance of the modified Probabilistic algorithm and are running the full version of the algorithm. Every other robot simply sends a message pair with 50% probability each time step.

We chose this type of population mix, so that we can have a uniform density of robots in the environment and be able to model the path of robots through an observer's communication range as a straight path. This was done to allow easier modeling of the association problem. The low number of mobile-mobile robot interactions is ignored in the modeling process. An example of a realistic multi-robot system with such a population mix can be seen in heterogeneous system composed of fast aerial and slow surface-based vehicles. For the experiments we ran for the mobile case we assumed that the robots' wireless and visual communication ranges are equal. The initial conjecture that varying the fractions of mobile robots would

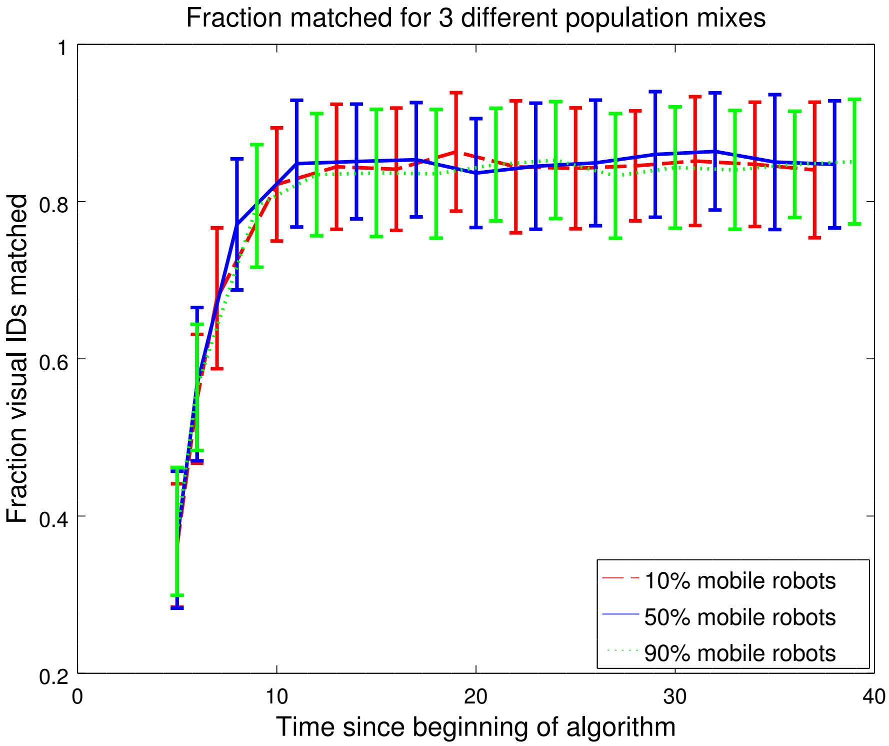


Figure 5.1: This figure shows the changes in the fraction matched metric with time for 3 different population mix values. All other parameters are equal to the default values. Red dashed line corresponds to 10% mobile robots, blue solid — to 50%, and green dotted — to 90%.

have a significant impact on the algorithm's performance was incorrect as can be seen from Fig. 5.1 and Fig. 5.2 which show simulation data for 3 different population mixes: 10%, 50%, 90% mobile robots. However, establishing why the conjecture is incorrect is not part of the scope of this work.

5.5 System Parameters and Their Effect on Algorithm Performance

In this section, we identify several key robot and environment parameters which affect the modified Probabilistic algorithm's expected performance. We examine, through simulation, their effect on the two performance metrics mentioned in the previous section.

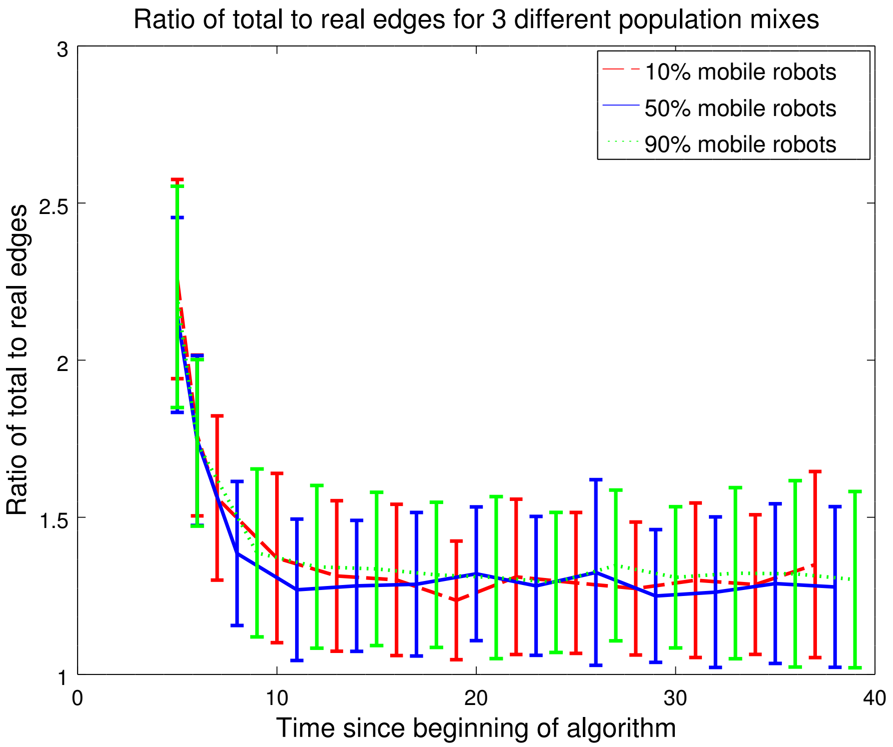


Figure 5.2: This figure shows the changes in the total to real edges metric with time for 3 different population mix values. All other parameters are equal to the default values. Red dashed line corresponds to 10% mobile robots, blue solid — to 50%, and green dotted — to 90%.

The key parameters have an effect on the two components of the mobile association problem. The multiplicative subprocess is affected by how much time the algorithm has had to operate on the edges and how many edges there are. The longer an observing robot can observe its neighborhood without any changes, the less ambiguity it will encounter. Conversely, the additive subprocess is affected by how many new IDs there are in range of a robot at every time step. As the number of new IDs increases the impact on how ambiguous the i-state is from the additive subprocess increases. The following is a list of key parameters.

1. Communication range size

2. Robot density in the environment

3. Robot speed through the environment

For our experimental settings, we hold all but one parameter unchanging and examine its effect by varying its value. The parameters are given in multiples of the robot diameter where applicable. We have chosen the following parameter values as default values which we use when a parameter is not varied:

- Robot speed — 1.4 robot diameters per time step

- Robot communication range radius — 16 robot diameters for both wireless and visual range

- Robot density in the environment — based on an average separation between robots' centers of 4 robot diameters

The algorithm displays two distinct phases: a **transient initialization** phase and a **steady-state** phase. The transient phase is a result of the initial steps introducing new IDs in the entire communication range, as the robots wake up and start running

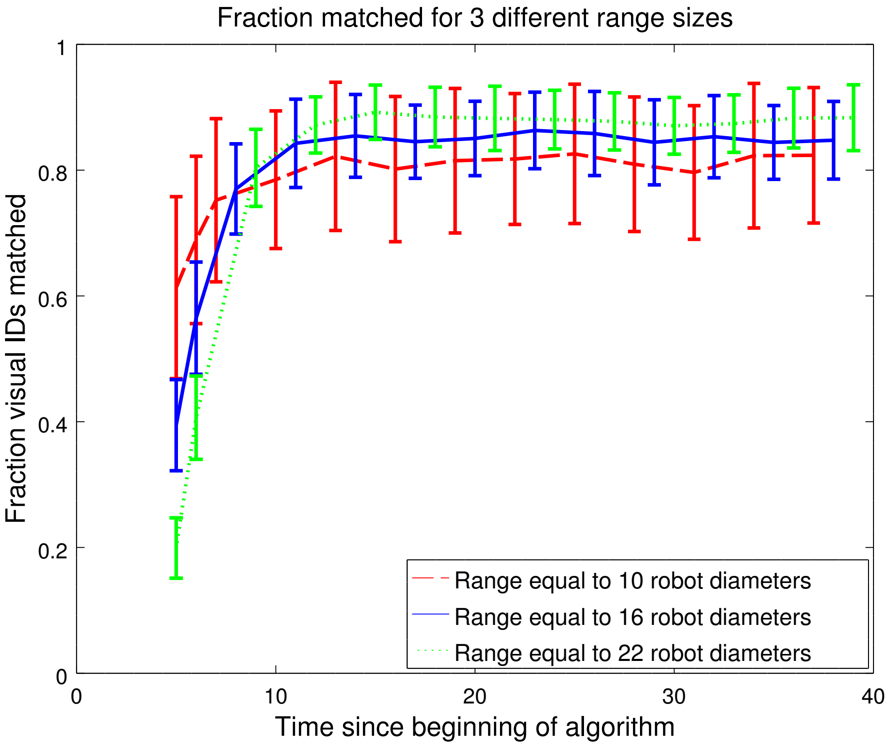


Figure 5.3: This figure shows the changes in the fraction matched metric with time for 3 different range sizes. Range is given in robot diameters. All other parameters are equal to the default values. Red dashed line corresponds to 10 diameters, blue solid — to 16, and green dotted — to 22.

the algorithm. The steady-state phase is what the algorithm's performance looks like once each robot has observed its neighborhood for a few time steps and the new IDs entering the i-state come only from a section at the front of the communication range.

Fig. 5.3 and Fig. 5.4 show the evolution of the two performance metrics for three different range sizes. The range sizes which have been plotted on the two figures are 10, 16 and 22 robot diameters.

Fig. 5.5 and Fig. 5.6 show the evolution of the two performance metrics when varying the robot density in the environment. The range radius for this set of exper-

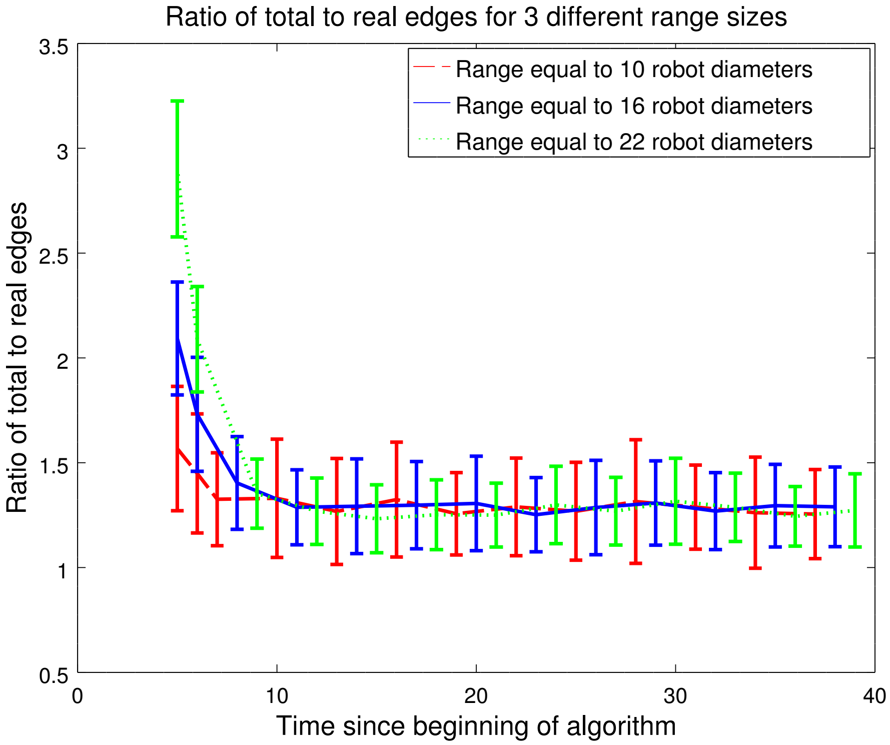


Figure 5.4: This figure shows the changes in the total to real edges metric with time for 3 different range sizes. Range is given in robot diameters. All other parameters are equal to the default values. Red dashed line corresponds to 10 diameters, blue solid — to 16, and green dotted — to 22.

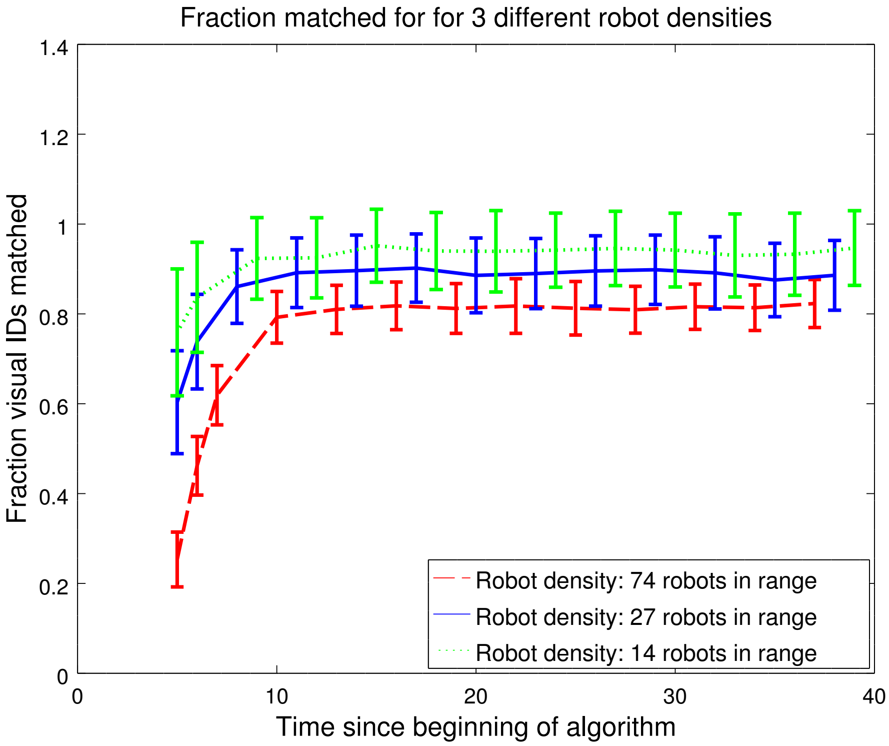


Figure 5.5: This figure shows the changes in the fraction matched metric with time for 3 different density values. Density is given by the resultant average number of robots withing communication range. All other parameters are equal to the default values. Red dashed line corresponds to 74 robots in range, blue solid — to 27, and green dotted — to 14.

iments is 20 robot diameters, which combined with the average separation between robots of 8, 5, and 3.2 robot diameters gives an average of 14, 27 and 74 robots in range for each density tested.

Fig. 5.7 and Fig. 5.8 show the evolution of the two performance metrics as robot speed is varied. The robot speeds plotted in the two figures are $\frac{1}{8}^{th}$, $\frac{1}{11}^{th}$, and $\frac{1}{20}^{th}$ of the communication radius per time step. Equivalently these values are 2, 1.4 and 1.25 robot diameters per time step.

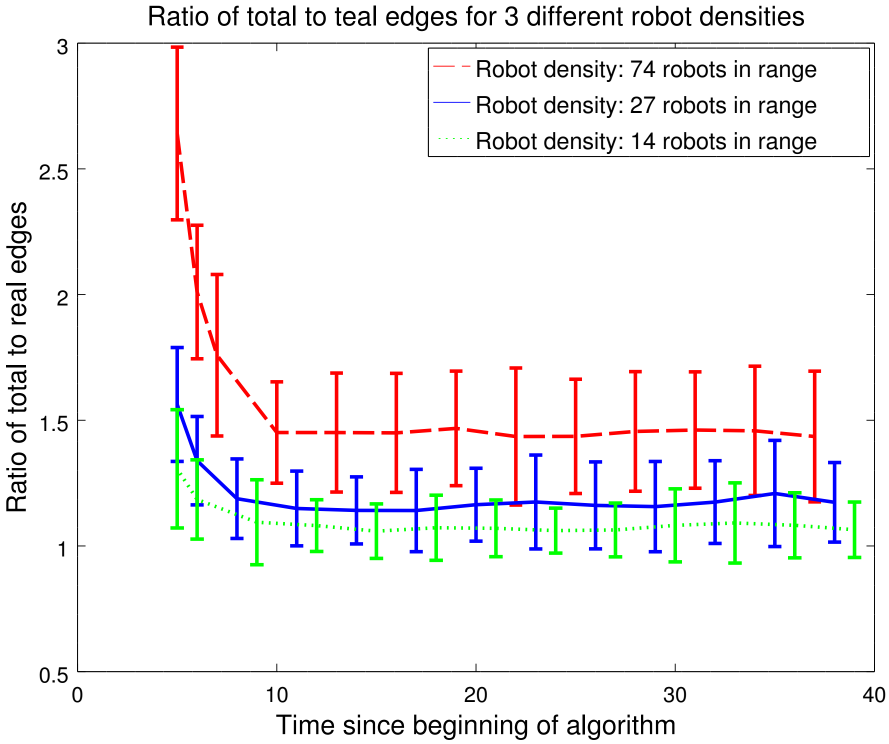


Figure 5.6: This figure shows the changes in the total to real edges metric with time for 3 different range sizes. Density is given by the resultant average number of robots withing communication range. All other parameters are equal to the default values. Red dashed line corresponds to 74 robots in range, blue solid — to 27, and green dotted — to 14.

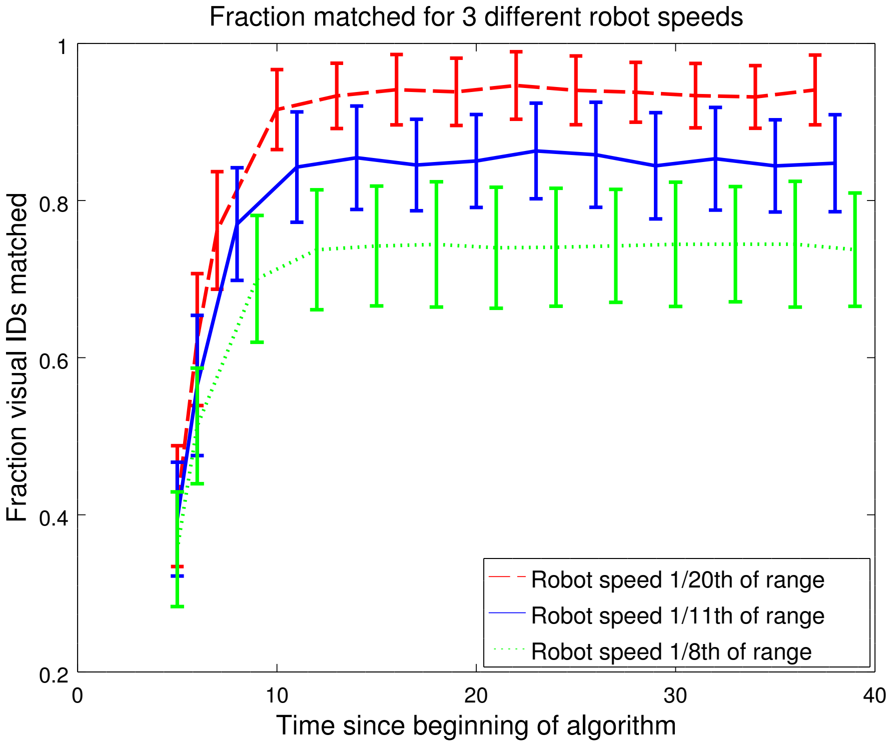


Figure 5.7: This figure shows the changes in the fraction matched metric with time for 3 different robot speeds. Robot speed is given in fractions of the communication range radius. All other parameters are equal to the default values. Red dashed line corresponds to 1/20th of range, blue solid — to 1/11th, and green dotted — to 1/8th.

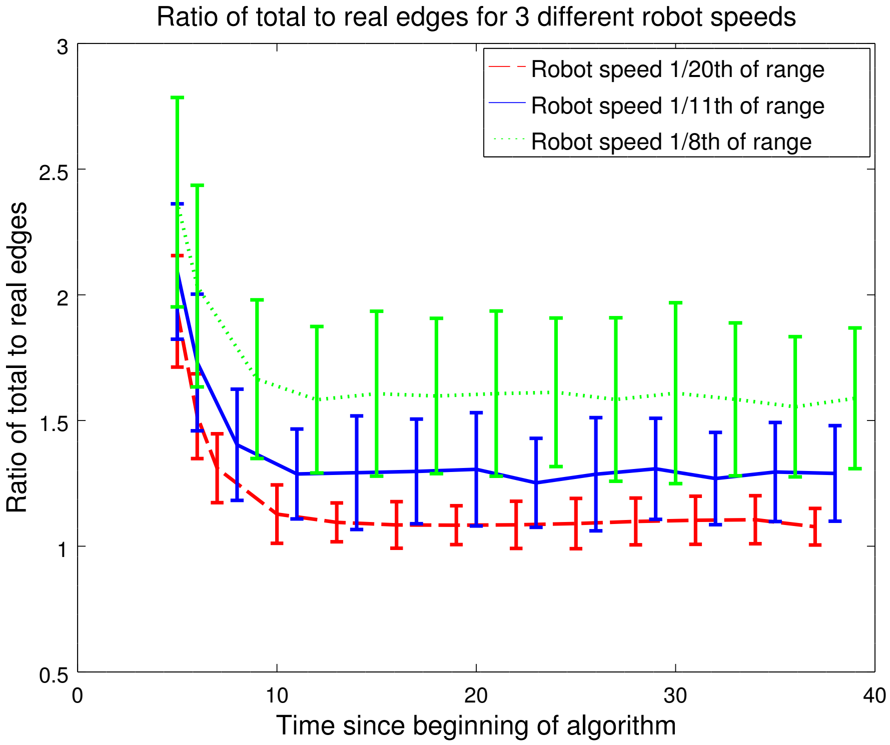


Figure 5.8: This figure shows the changes in the total to real edges metric with time for 3 different robot speeds. Robot speed is given in fractions of the communication range radius. All other parameters are equal to the default values. Red dashed line corresponds to 1/20th of range, blue solid — to 1/11th, and green dotted — to 1/8th.

5.6 Macroscopic Model of the Mobile Case

In this section we examine the macroscopic model for predicting the average expected behavior of a robot in a multi-robot system running the modified Probabilistic algorithm.

The model will be restricted to the case where the wireless range is equal to the visual range. At this stage only this somewhat restricted version of the mobile association problem is understood well enough by the author to be modeled correctly. Further work is needed to model the case where the wireless range is larger than the visual range.

Restricting the communication range ratio to 1 removes the interaction between old and new IDs in the i-state. In effect the IDs which enter the i-state of an observing robot at the same time form an independent smaller i-state. If we allow the wireless range to be larger than the visual, then the wireless ID corresponding to a robot may enter the i-state of an observer several time steps before its corresponding visual ID. This necessitates the addition of edges between all new visual and old but yet unassociated wireless IDs.

5.6.1 Underlying Processes for the Mobile Case

The robots we model have circular communication ranges centered around them. Each time step the robot moves a particular distance d, which changes the area which is covered by the communication range. If the distance is less than the diameter of the range, there will be overlap between the areas covered by the communication range in two consecutive time steps. See Fig. 5.9 for a graphic representation of the overlap. This overlap can be computed using Eq. (5.2). The area newly covered by the communication range, which we call the **frontier** can then be computed simply by subtracting the overlap area from the full circle area. The area which was covered

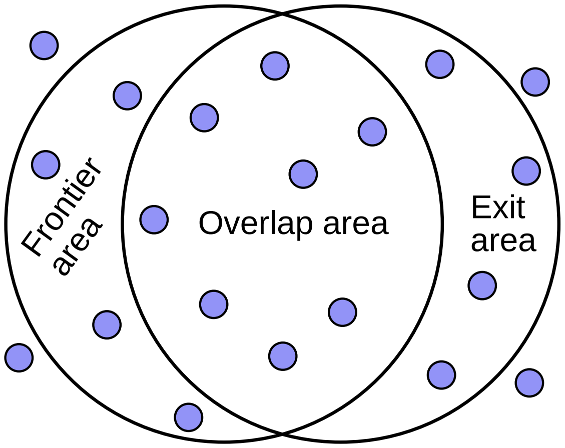


Figure 5.9: Schematic of overlap between two circles separated by a small distance. The frontier corresponds to a section of the environment which is newly covered by the communication range. On the opposite side from the frontier is the exit area. This is a section of the environment which left the communication range at the end of the current time step. The smaller circles correspond to robots randomly positioned in the environment.

in the previous time step but not in the current is called the **exit** area. It has the same size as the frontier. This area computation assumes the robot is moving in a straight line and does not change direction.

$$A_O = 2r^2 \cos^{-1} \frac{d}{2r} - d\sqrt{r^2 - (\frac{d}{2})^2} \tag{5.2}$$

As the robot continues on its path, the area its communication range covers keeps changing. Previously covered area moves closer to the back end of the communication range. If the distance covered each time step is small, then there will be several regions in the communication range, which would have spent different time *in range* of the observing robot. These areas can be seen in Fig. 5.10. We call these areas, **age regions**, as each has a corresponding number which represents how many time steps it has been in range of the observing robot. The age of the regions tells us how

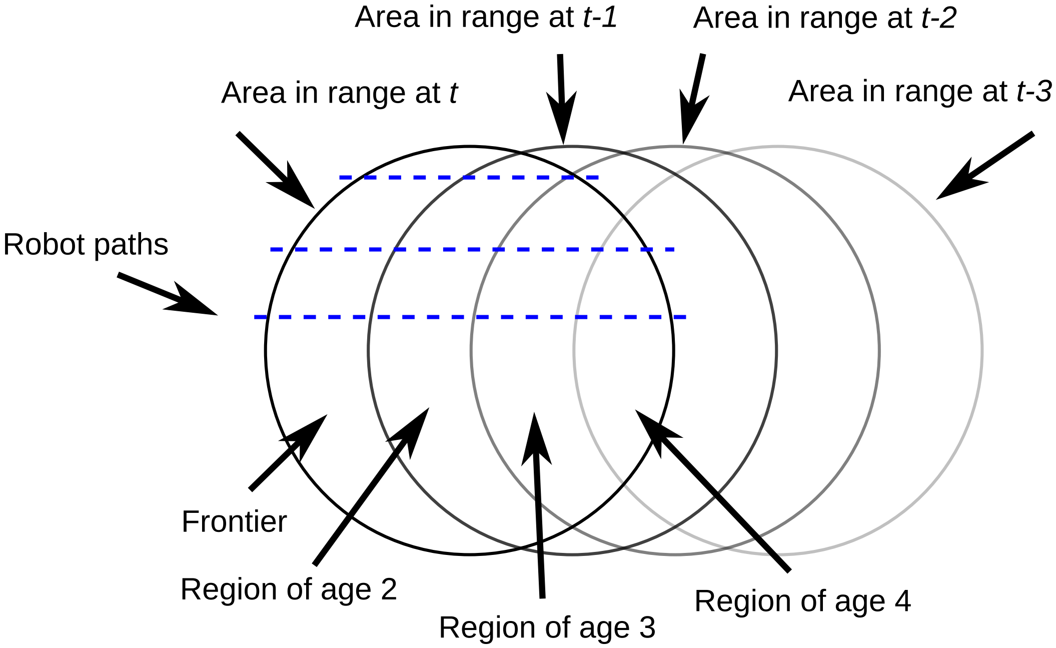


Figure 5.10: Geometry of the overlap between the area covered by the communication range of a mobile robot at different time steps and the resulting distribution of regions by age in the current area covered by the communication range. The blue dashed lines show the varying lenght of the paths of robots.

how many times the algorithm has been applied.

The area of the age regions can be computed by applying Eq. (5.2) with different multiples of d to find each area. The area and the expected robot density in the environment can be used to compute the expected population of robots in each region. This in turn allows the computation of the mean number of IDs which will enter each time step through the frontier region. We call the process of new IDs entering the communication range **positive migration**. The positive migration determines the size of the additive subprocess in the association problem. The IDs which exit the communication range determine the **negative migration**.

5.7 Fraction Matched

We have applied the stationary case fraction matched model to the mobile case when the two communication ranges are equal. As mentioned earlier, each region constitutes an independent i-state and its fraction of unambiguously associated visual IDs can be approximated using the *fraction matched* model. To get an idea of how many matched visual IDs there are in total in the full range, we just need to compute the fraction of visual IDs which have been unambiguously associated for each age region and then sum the results. Figures 5.11, 5.12, and 5.13 show how the predictions made by the *fraction matched* model compare with the results of the simulated trials.

It is evident that there is some inaccuracy in applying the *fraction matched* model to the mobile case of the association problem. One possible contribution to the inaccuracy is that the actual numbers of edges which enter the i-state at different time steps have some distribution around the mean, which affects the fraction visual IDs which get unambiguously associated. Another possible contribution is the loss of IDs from the i-state as the age region to which they belong shrinks. Further work is needed to establish why the *fraction matched* model underestimates by such a degree.

5.8 Ratio of Total to Real Edges

5.8.1 Extending the Branching Processes Model

To predict the expected ratio of total to real edges, we are going to extend the Branching Processes model from chapter 4 to handle the additional additive subprocess. To achieve this we need to incorporate both the positive and negative migration in the model. The Branching Processes literature offers ways of handling migration as well.

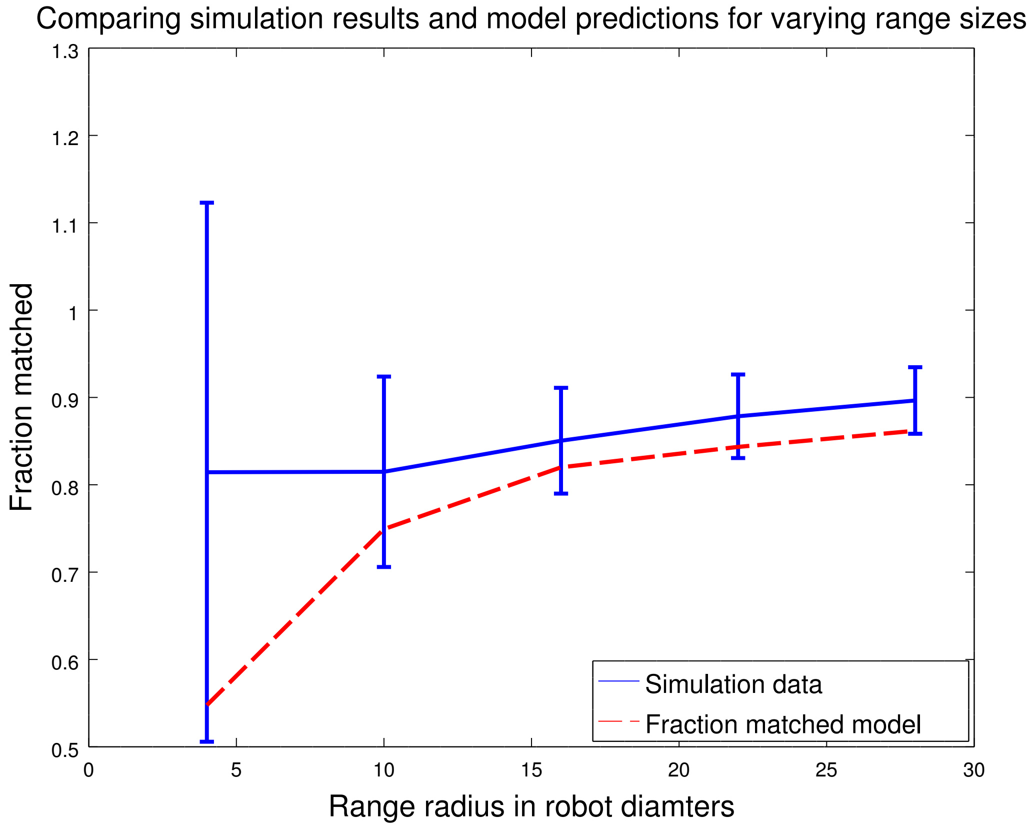


Figure 5.11: This figure shows a comparison between simulation data for several range size values and the fraction matched model (in red, dashed line) prediction.

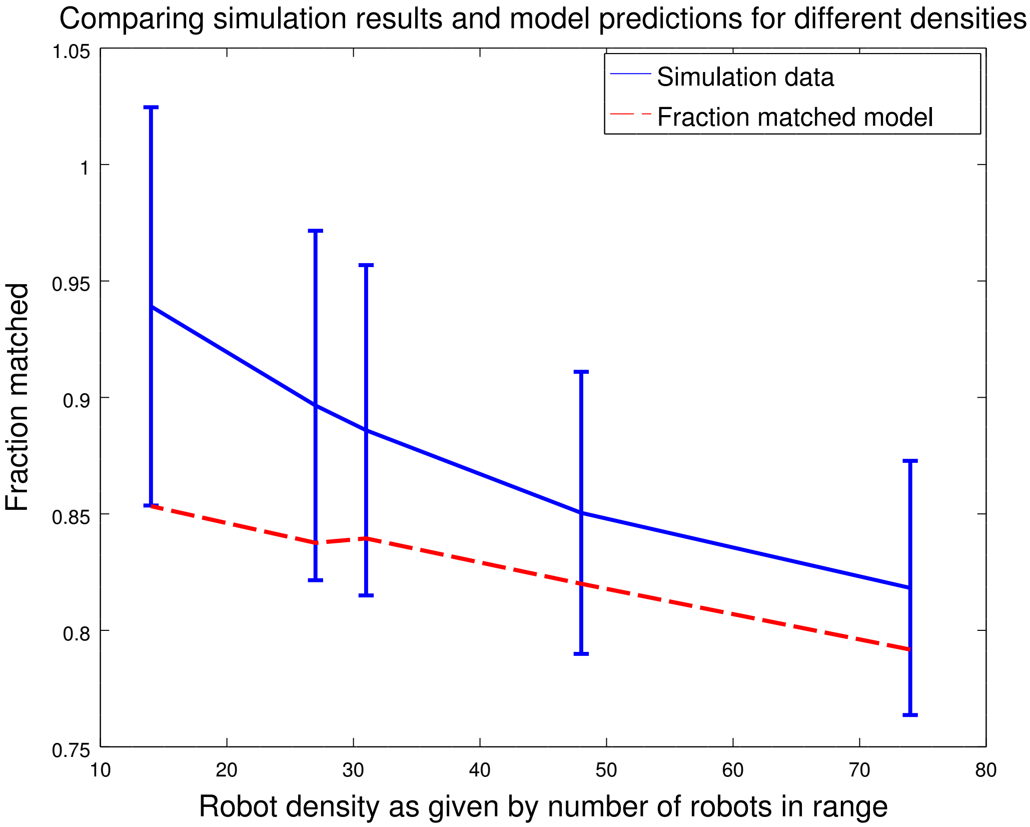


Figure 5.12: This figure shows a comparison between simulation data for several density values and the fraction matched model (in red, dashed line) prediction.

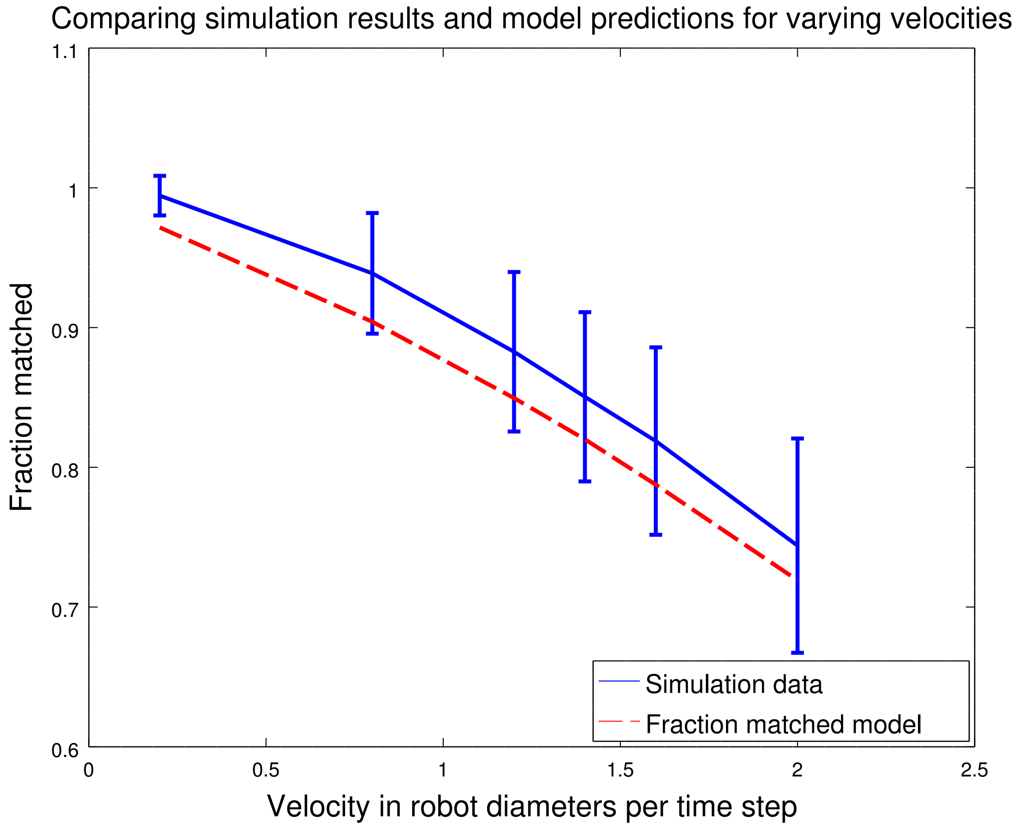


Figure 5.13: This figure shows a comparison between simulation data for several velocity values and the fraction matched model (in red, dashed line) prediction.

As in the stationary case, we need to take into account the mean reproduction rate of the two types of edges we have. These will be handled by the variables m_F and m_R. In the mobile case, the values of the mean reproduction rate are no longer 0.5 and 1 for the fictitious and real edges, respectively. Now we need to add the contribution due to negative migration. This will be computed as a correction factor to the mean reproduction rates based on how IDs exit the communication range. The two negative migration factors will be denoted with p_{F-} and p_{R-}, which are the probability of an edge *migrating out*. This is determined by the probability of one or both of the IDs composing the edge moving out of communication range.

In addition to the reproduction rate of the individual edges, we also need to incorporate the positive migration of edges. These will be denoted with λ_{F+} and λ_{R+} for positive migration of fictitious and real edges.

The above parameters can be used to construct the **mean matrix** M of the branching process, Eq. (5.3). This matrix can be used to compute the expected number of individuals of each type given a starting population P_i and a number of generations (time steps) t, Eq. (5.4).

The mean matrix is constructed as follows:

$$M = \begin{bmatrix} 1 & \lambda_{R+} & \lambda_{F+} \\ 0 & m_R(1-p_{R-}) & 0 \\ 0 & 0 & m_F(1-p_{F-}) \end{bmatrix} \tag{5.3}$$

Here a population vector is defined as follows:

$$P_i' = \begin{bmatrix} 1 & e_R & e_F \end{bmatrix} \tag{5.4}$$

where:

- the first element in the matrix and population vector stands for a **migrator individual**, which is a mathematical tool used to introduce the additive positive migration
- e_R is the number of real edges in the i-state
- e_F is the number of fictitious edges in the i-state
- i is the time step during which the population was counted

The mean matrix is applied to the initial population to get a resulting population after t time steps as follows:

$$P_{i+t} = M^t \times P_i \tag{5.5}$$

The Branching Processes literature also offers an approximation for the expected number of individuals of each type if the mean migration and reproduction rates remain constant. The approximation for a single individual type with $m < 1$ is summarized in the following equation:

$$E[P_t] = \frac{\lambda}{1 - m} \tag{5.6}$$

5.8.2 Converting System Parameters to Branching Processes Parameters

In this section we expand upon the construction of two specific Branching Processes models and how we convert the robot and environment parameters into Branching Processes parameters. In an earlier section we explained how the motion of a robot through the environment creates areas inside the communication range which have spent different times *in range* of an observing robot. We specifically distinguish the frontier region. In this section we are going to describe two BP models built around the idea of different treatment of the remaining age regions.

The first model ignores the distinction between age regions and treats all areas which are not the frontier as one region - we call this region the **rest**. The second model explicitly models each age region and the transition of robots and IDs from one region to the next. We call the first model the **two-region model** and the second the **multi-region model**.

In the two-region model we get a mean matrix as described in Eq. (5.3) of size 3 by 3. The mean positive migration for each edge type is calculated using the area of the frontier region and the mean density of robots in the environment. For the fictitious edges $\lambda_{F+} = w_{\text{new}} v_{\text{new}} - v_{\text{new}}$ and for the real edges $\lambda_{R+} = v_{\text{new}}$. In order for the model to give accurate results, we need to take into account the variance of w_{new} and v_{new}. In the case of equal wireless to visual range the two numbers are equal and their variance is identical. So, the value of the expected positive migration of fictitious edges, after accounting for variance, becomes:

$$\lambda_{F+} = \text{Var}[v_{\text{new}}] + E[v_{\text{new}}]^2 - E[v_{\text{new}}] \tag{5.7}$$

We gathered data from the simulation for the variance of the expected number of new visual and wireless IDs. This can also be done by modeling the spatial variance in the position of the robots in the environment. We did not attempt such modeling in this work.

The negative migration probability is calculated using the following formula:

$$p_{R-} = p_{F-} = \frac{v_{\text{exit}}}{v_{\text{rest}}} + \frac{w_{\text{exit}}}{w_{\text{rest}}} - \frac{v_{\text{exit}} w_{\text{exit}}}{v_{\text{rest}} w_{\text{rest}}} \tag{5.8}$$

This is motivated by the fact that each edge in the *rest* area will migrate out if one or both of its IDs migrate out. There are v_{rest} IDs of each type in the *rest* area and on average about v_{exit} IDs of each type exit the range. The number of exiting IDs is computed similarly to how the population of frontier IDs is computed — the area of the region which will exit the communication range in the next time step has the same size as the frontier. So, we get a probability of exiting the *rest* area which is given by the mean number of IDs exiting the range over the mean number of IDs in the *rest* area.

In the multi-region model we explicitly account for the movement of IDs and edges through each region. As can be seen in Fig. 5.10, the size of each age region gets smaller as its age increases. Since fictitious and real edges are independent from each other, we can split the mean matrix into one dedicated for each edge type. In addition, the mean matrix now has a larger size — equal to the total number of age regions plus 1 (for the migrator). Now we get a correction factor for the mean reproduction rate based on how the area of two adjacent region compares. For example, the mean reproduction rate for edges going from an age region with an area $A_i = 10$ to an age region with an area $A_j = 9$ will need to be adjusted by a factor of $\frac{A_j}{A_i} = 0.9$. The positive migration for the multi-region model is still accounted for by the number of new IDs in the frontier region. The positive migration is added only to the frontier region's count of edges. At each time step, the surviving edges (those which were not eliminated by observation and those whose IDs remained in range) of each age region are moved to the next age region. All the edges in the last age region exit the i-state once the robot moves.

An example mean matrix for fictitious edges in the multi-region model can be seen in Eq. **??**. The fictitious edge population of the communication is defined as in Eq. 5.10. A similar matrix and population can be created for the real edges.

$$M = \begin{bmatrix} 1 & \lambda_{F+} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{A_2}{A_1}m_{F-} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{A_3}{A_2}m_{F-} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{A_4}{A_3}m_{F-} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{A_5}{A_4}m_{F-} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.9}$$

$$P'_i = \begin{bmatrix} 1 & e_{A1} & e_{A2} & e_{A3} & e_{A4} & e_{A5} \end{bmatrix} \tag{5.10}$$

When computing the steady-state results for the performance metrics we did not use the approximations from the Branching Processes literature, instead we chose a starting population and applied the mean matrix to it for a predetermined large number of time steps. We then used the values to which the population tends with time as our steady-state prediction. These match well with the approximation results.

Figures 5.14, 5.15, and 5.16 show the predictions of the two models (green for the two-region model, and blue for the multi-region model) and how they compare to the data gathered from simulation. Both models track the simulation mean closely and fall well withing 1 standard deviation from the mean.

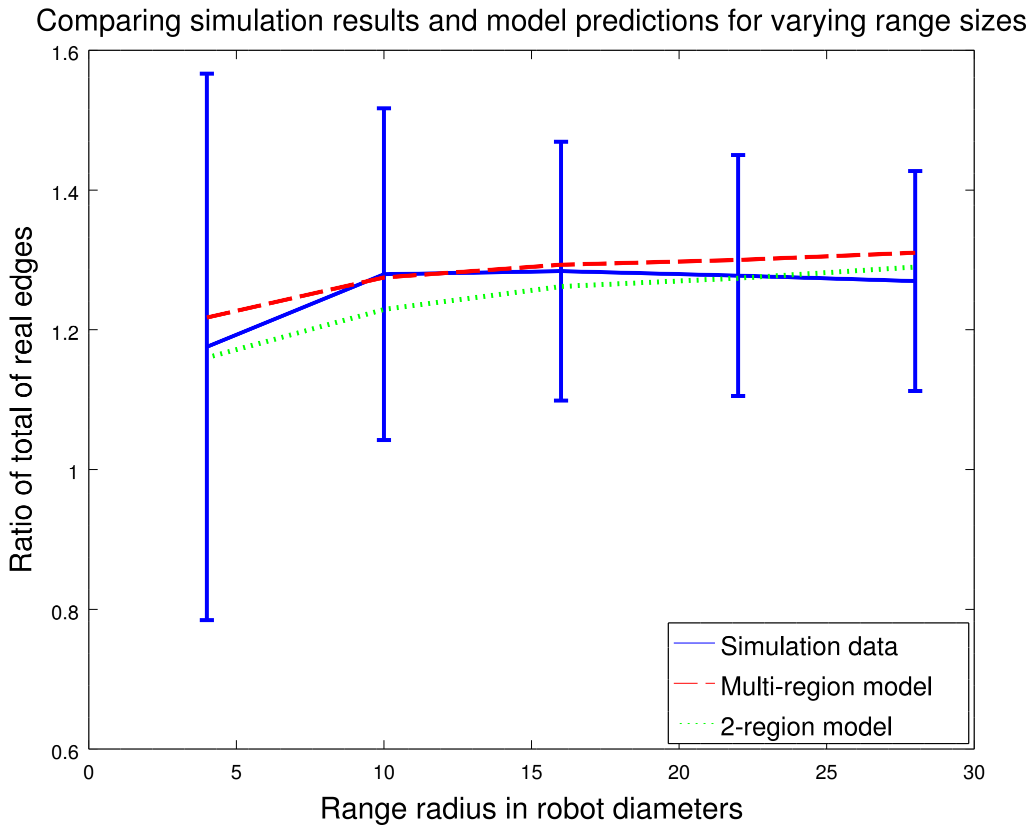


Figure 5.14: This figure shows a comparison between simulation data for several range size values and 2-region model (in green, dotted line) and multi-region model (in blue, dashed line) predictions.

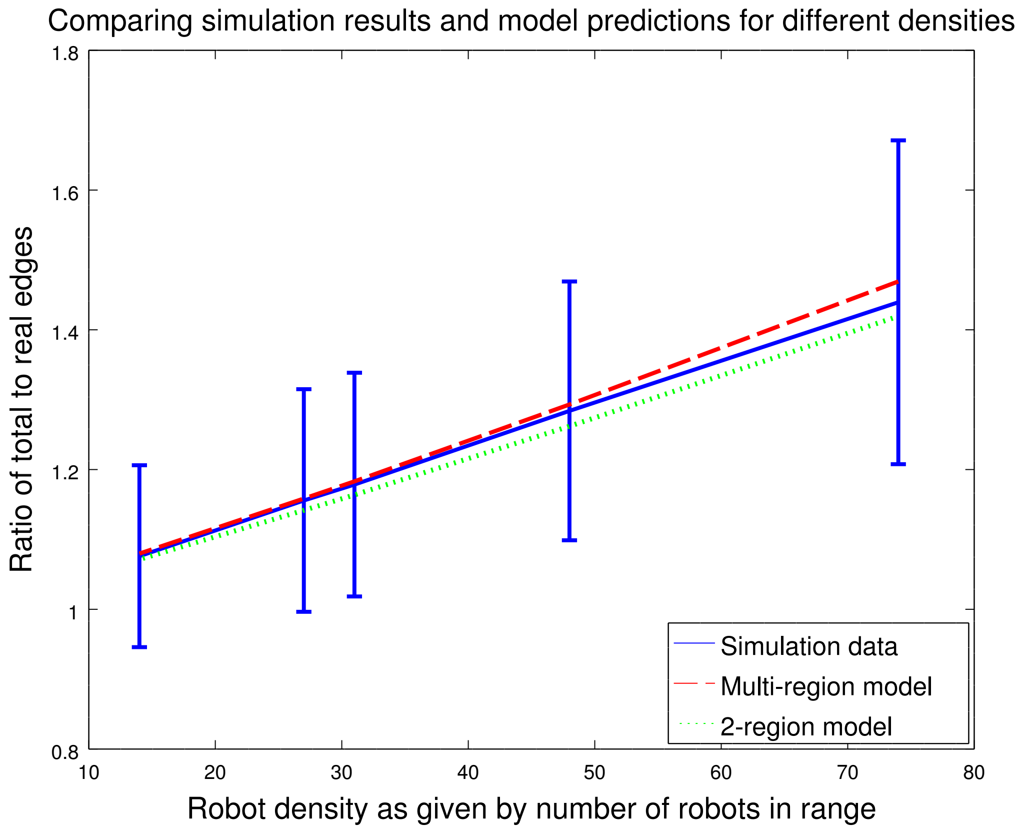


Figure 5.15: This figure shows a comparison between simulation data for several density values and 2-region model (in green, dotted line) and multi-region model (in blue, dashed line) predictions.

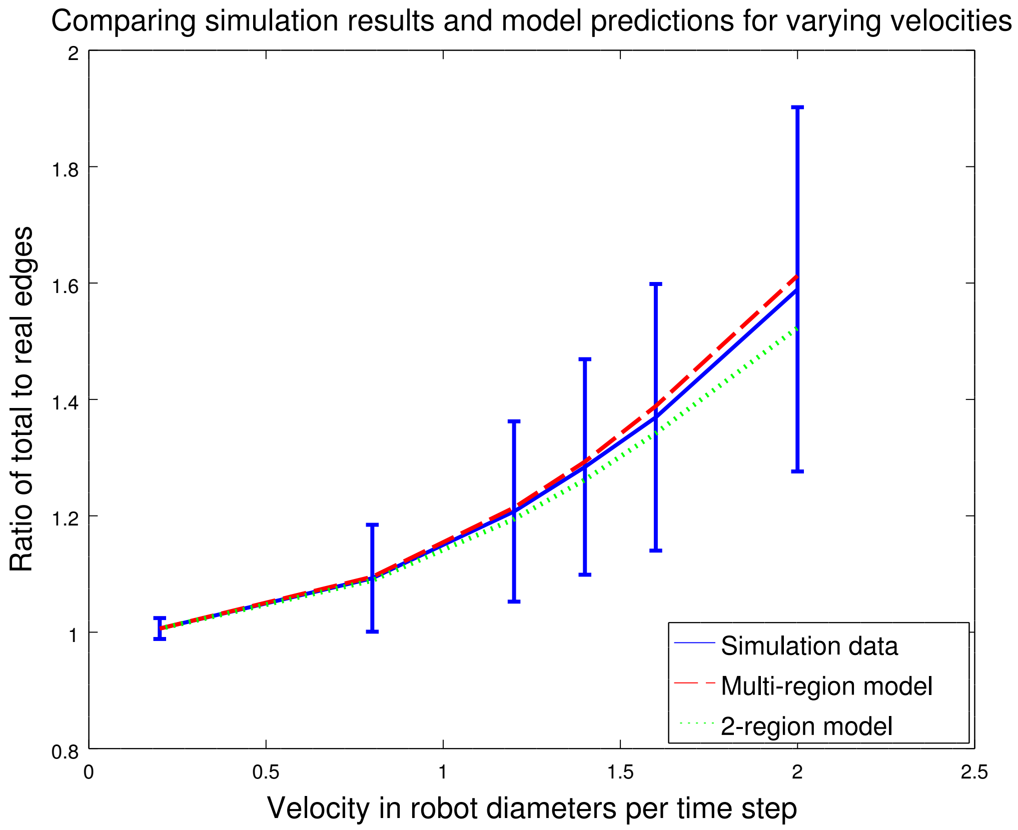


Figure 5.16: This figure shows a comparison between simulation data for several robot velocity values and 2-region model (in green, dotted line) and multi-region model (in blue, dashed line) predictions.

6. FUTURE WORK AND CONCLUSION

6.1 Future Work

6.1.1 Larger Wireless Range Than Visual Range

In this work we only examine the case of equal wireless and visual communication ranges. This allows the algorithm to take advantage of properties which exist only in this restricted version of the association problem. More specifically, wireless and visual IDs belonging to the same robot enter the i-state during the same time step. When the assumption of equal ranges is relaxed, a wireless ID may enter the i-state much sooner than the corresponding visual ID. This causes IDs from different age regions to interact and form edges — an edge must be added between all active unassociated wireless IDs and newly discovered visual IDs. The Branching Processes model can still handle the prediction of the total to real edges metric, however, it relies on an accurate model of how new edges are added. Further work is needed to build such a model. The fraction matched model will also need to be updated to handle the lack of independence between age regions.

6.1.2 Different Population Mixes

The results from simulations we ran show that varying the ratio of stationary to mobile robots in the environment had no significant impact on the performance of the algorithm. This applies to both the steady state mean and variance. Stationary robots travel through the communication range of a mobile robot in parallel lines along the direction of motion of the observer. The trajectories of mobile robots through the communication range of an observer are much more varied — they are no longer parallel to the direction of motion and can enter and exit the range from a

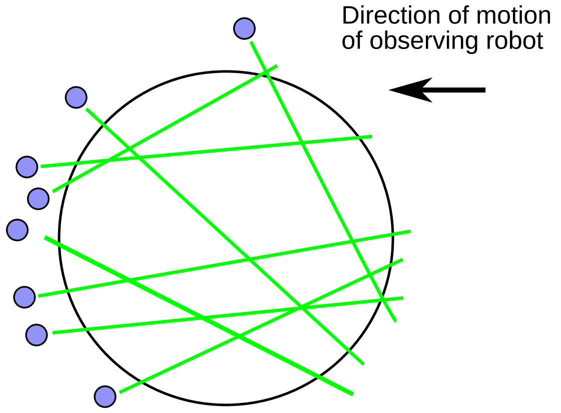


Figure 6.1: A diagram of potential trajectories of mobile robots passing through the communication range of a mobile observing robot. The smaller circles correspond to robots moving randomly through the environment and the greed lines denote their potential trajectories. The length of these trajectories no longer depends simply on the distance between the robot and the line of motion of the observer.

wider area than just the frontier and exit regions. A diagram can be seen in Fig. 6.1. In addition, the time a mobile robot spends in range of an observer now depends on the vector sum of their speeds. The different geometry of the paths of robots through the communication range of an observer for mobile robots produces similar results as the stationary robots. Developing an understanding of why the ratio of stationary to mobile robots does not affect algorithm performance is a direction for future work.

6.1.3 Noise and Errors in Communication

The communication of the multi-robot system in this work is assumed to be ideal *i.e.* there is no noise, dropped messages or errors. Addressing a more realistic communication system is a major direction for future work. The observations made by robots about their neighborhoods will now have some degree of uncertainty. The edges corresponding to potential associations can no longer be eliminated based on a single or even multiple observations.

In the case of ideal communication, a simple recording of how often an edge, with both of its IDs active, is seen in the observation matrix results in a very clear distinction between real and fictitious edges. As time passes and more observations are collected, real edges will tend towards being active during 50% of the time while fictitious edges will tend towards 25%. This is based on the probability of an ID to be active at any given time and these values will be different if that probability is not 50%. The IDs of real edges are directly correlated in their activations, while the IDs of fictitious edges are independent. The same principle can be applied to unreliable observations. Furthermore, an understanding of the distribution of false positives or false negatives in correlated activations can be used to inform the Branching Processes model and the fraction matched model. Further work is needed to apply the analysis and models presented in this work to systems with unreliable communication.

6.2 Conclusion

Robots in multi-robot systems often need to address conveniently located neighbors. A common approach to achieve this is to give the robots external markers which are paired to a network address. This work identified and provided a detailed formal description of the communication channel association problem in which a robot algorithmically associates its neighbors to their network addresses without the use of external markers or global information. The probabilistic algorithm used for solving the association problem uses visually identifiable gestures (such as a light being turned on or off) and wireless messages to quickly establish the one-to-one association between the wireless IDs of neighboring robots and their locally assigned visual IDs. The gestures along with the sensors used to detect them form a situated communication channel. The approach relies on the physically situated property of this channel to create a meaningful to the sensing robot association between neighboring robots and their wireless IDs. We identified key system parameters — density of robots in the environment, communication range, and robot speed — and explored their effect on the performance of the probabilistic algorithm through simulations of a multi-robot system employing the algorithm. We also discovered that a fourth system property — ratio of stationary to mobile robots in the environment — surprisingly has no significant effect on algorithm performance. We demonstrated that a common modeling approach for similar problems, absorbing Markov chain, is too expensive to apply to realistic instances of the association problem. We created a set of models based partly on Branching Processes which are used to predict the macroscopic behavior of a multi-robot system running our algorithm. The models are validated using the simulation results. These models can be used to evaluate the feasibility of using the probabilistic algorithm to solve the association problem for a multi-robot system with given parameters or during the process of designing a system to achieve some desired threshold performance values.

REFERENCES

[1] Philip E. Agre and David Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*, AAAI'87, pages 268–272. AAAI Press, 1987.

[2] D. Berend and T. Tassa. Improved bounds on Bell numbers and on moments of sums of random variables. *Probability and Mathematical Statistics*, 30(2):185–205, 2010.

[3] Yoann Dieudonné, Shlomi Dolev, Franck Petit, and Michael Segal. Deaf, dumb, and chatting asynchronous robots. In *Proceedings of the 13th International Conference on Principles of Distributed Systems*, OPODIS '09, pages 71–85, Berlin, Heidelberg, 2009. Springer-Verlag.

[4] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, 2000.

[5] A. Franchi, G. Oriolo, and P. Stegagno. Mutual localization in a multi-robot system with anonymous relative position measures. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3974–3980, Oct 2009.

[6] S. Garrido-Jurado, R. Muñoz Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recogn.*, 47(6):2280–2292, June 2014.

[7] Charles M. Grinstead and J. Laurie Snell. *Introduction to Probability*, chapter 11: Markov Chains. American Mathematical Society, 1997.

[8] Alvaro Gutirrez, Alexandre Campo, Marco Dorigo, Daniel Amor, Luis Mag-

dalena, and Flix Monasterio-Huelin. An open localization and local communication embodied sensor. *Sensors*, 8(11):7545, 2008.

[9] P Haccou, P Jagers, and V.A. Vatutin. *Branching Processes: Variation, Growth, and Extinction of Populations.* Cambridge Studies in Adaptive Dynamics, 2007.

[10] Andrew Howard, Lynne E. Parker, and Gaurav S. Sukhatme. The sdr experience: Experiments with a large-scale heterogenous mobile robot team. In *9th International Symposium on Experimental Robotics 2004*, Singapore, Jun 2004.

[11] P. Ivanov and D. A. Shell. Associating nearby robots to their voices. In *Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALife)*, 2014.

[12] Steven M. LaValle. Tutorial: Filtering and planning in information space. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS'09, pages 1–1, Piscataway, NJ, USA, 2009. IEEE Press.

[13] Yves Lespérance and Hector J. Levesque. Indexical knowledge and robot action - a logical account. *Artificial Intelligence*, 73:69–115, 1994.

[14] N. Mathews, A.L. Christensen, R. O'Grady, and M. Dorigo. Spatially targeted communication and self-assembly. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2678–2679, Oct 2012.

[15] Nithin Mathews, Gabriele Valentini, AndersLyhne Christensen, Rehan OGrady, Arne Brutschy, and Marco Dorigo. Spatially targeted communication in decentralized multirobot systems. *Autonomous Robots*, pages 1–19, 2015.

[16] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE, May 2011.

[17] Kasper Støy. Using situated communication in distributed autonomous mobile

robotics. In *In Proceedings of the 7th Scandinavian conf. on Artificial Intelligence*, 2001.