TOPOLOGY CREATION AND IMPACT ON CONSENSUS IN MULTI-AGENT SYSTEMS

A Dissertation by

Hicham Hatime

M.S. Electrical Engineering, Wichita State University, 2001

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

May 2013

TOPOLOGY CREATION AND IMPACT ON CONSENSUS IN MULTI-AGENT SYSTEMS

The following faculty members have examined the final copy of this dissertation form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of doctorate of philosophy with a major in Electrical Engineering.

_____
John M. Watkins, Committee Chair

_____
Ravi Pendse, Committee Member

_____
Kameswara Rao Namuduri, Committee Member

_____
Driessen Brian, Committee Member

_____
Vinod Namboodiri, Committee Member

Accepted for the College of Engineering

_____
Zulma Toro-Ramos, Dean

Accepted for the Graduate school

_____
Abu S. M. Masud, Interim Dean

DEDICATION

To my family for their support

ACKNOWLEDGEMENTS

I would like to thank all, who aided in one way or another in the realization of this dissertation. The support, recommendations and guidance provided helped greatly in accomplishing and reaching my goals.

I also wish to thank my family for their encouragements all along these long years, and all my friends who showed interest in my research progress.

ABSTRACT

Mobile ad hoc sensor networks are characterized by dynamic changes in communication links and network topology mainly due to node mobility. In such networks, it is a challenge to build a communication system that lasts longer and requires less reconfiguration and less communication overhead. In this study, we propose an on-demand topology reconfiguration approach for multi-agent systems aimed at enhancing the connectivity and performance. The proposed strategy groups nodes, elects cluster-heads and finally selects gateways for inter-cluster communication. This study, also, seeks to investigate the impact of topology structural characteristics on consensus building among multi-agent nodes. The structural properties evaluated are the algebraic connectivity ratio, average path, average cluster coefficient, average matching index, modularity and the average participation. The consensus is measured by nodes average state update. Statistical methods are employed to explore the interaction between these structural properties and the consensus. A theoretical analysis is provided to support the statistical results. Finally, a comparative study of three distributed strategies for task allocation in a multi-agent system is presented. The objective is to determine for each node its course of action and the tasks it needs to accomplish. The methods are based on self-organizing map technique, Hungarian method and a linear programming optimization formulation. A theoretical section is provided to support the dynamics of these techniques and some of the results.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

Chapter Page

TABLE OF CONTENTS (CONTINUED)

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

Figure Page

CHAPTER 1

INTRODUCTION AND PROBLEM STATEMENT

1.1 Introduction

Mobile ad hoc sensor networks are characterized by frequent topology updates and self-configuration. Networks are created spontaneously whenever nodes are within the transmission range of each other. The arrival and departure of nodes in a network is an on-going dynamic process due to nodes high mobility. This dynamic nature causes the nodes in mobile ad hoc sensor networks to lose connectivity quite frequently. Strategies are required to keep the network connected and adaptable to frequent changes in the network. The connectivity and adaptability are very crucial for a lasting communication. They require the nodes to be aware of their own state and the state of their surroundings nodes. For a randomly deployed mobile sensor, there are controllable and uncontrollable state parameters. The set of controllable parameters include power, transmission range, state, sensors, et cetera. The uncontrollable parameters are node position, neighbors, path, et cetera. The exceptions such as link failure and break downs are not part of the uncontrollable parameters. A condition for a connectivity and adaptability strategy is to encompass and balance between the two set of parameters. Neglecting or favoring one set over the other, generally results in a weak strategy. The third chapter in this research proposes a new periodic and on-demand topology reconfiguration strategy for a mobile ad hoc sensor network. This strategy combines a few controllable and uncontrollable parameters and generates a communication topology for randomly placed nodes.

Another issue related to mobile ad hoc sensor networks is collaboration. In many fields, multi-agent systems, in general, are often characterized by the complexity of the interaction between their elements. In biology, for example, the collective behavior of cells and tissues is a direct consequence of the properties of their placements and molecular constituents. In computer networks, the overall performance is measured by the network topology and each station's throughput, security, scalability of protocol, et cetera. In manufacturing, the quality of a final product is evaluated by its design, functionality, reliability and maintainability. In general, in any design, the placement of system constituents affects the parts interaction. In other words, interaction is affected by the topology. The fourth chapter in this research investigates the effect of structural properties of a topology on consensus building. The study involves measuring a few structural metrics for any given topology and defining a consensus or synchronization expression. Statistical techniques such as regression analysis and factor analysis can then be applied to show the interactions and interdependencies among the topological characteristics, and the impact of these structural properties on consensus. To support the statistical findings, a theoretical analysis is provided.

Another issue associated with multi-agent systems is task allocation. To reach an objective, the set of nodes, in a mobile ad hoc network have to coordinate in their endeavor. The process requires a communication topology and involves sharing information to reach a consensus. The overall task is then accomplished when each node accomplishes its scope of tasks. The key challenges in accomplishing a task in a multi-agent environment are task allocation and coordination. When fielding multiple nodes in an unknown area, the problem of uncertainty and unpredictability arises. In a dynamically changing environment, the problem becomes more complex to tackle. The fundamental problem is as follows, given a set of nodes

and targets (tasks) spread in a geographic area, how can a node decide on its course of action and contribute to the accomplishment of the overall task? The fifth chapter in this research proposes to compare three distributed techniques aimed at providing each node with autonomous decision making. These task allocation strategies are mutually exclusive and do not require any coordination among nodes. The study focuses on the cost of task allocation, fairness and workload.

In summary, chapter two provides a literature survey of the subjects. Chapter three builds a communication topology. Chapter four investigates the structural properties of the built topology. Chapter five compares the three node task assignment techniques.

1.2 Problem Statement

Topology control has been an interesting subject of research lately. Many areas such as chemistry, biology and ad hoc networks benefited from its development. Topology control can be divided into two categories: topology construction for building the initial topology and topology maintenance aimed at preserving the topology. In the area of mobile sensor ad hoc networks, topology control is faced with many challenges such as energy conservation, time-varying topology and low transmission quality. Many of the approaches in the literatures for communication topologies are either power mechanism based (e.g. energy cost) or hierarchical based (e.g. neighbors). There are no attempts to combine parameters of the power and hierarchical techniques in a new approach that identifies the available clusters, elect cluster heads, select gateways and establish communication links among all nodes.

Constructing a communication topology for randomly dispersed nodes is necessary for exchanging information. The initial scope of information available to each node, in a mobile

sensor network, is usually limited by the information collected. Increasing this scope requires either collecting more data or requesting it from other nodes. The process of exchanging data among nodes in order to improve their partial scope is a step in the consensus building path. There are many factors that influence the consensus building process. Communication topology is an important factor in the equation. The relation between the communication topology and the consensus is still a black box. No study has been performed to evaluate the impact of a communication topology on the consensus building.

The concept of information exchange and consensus building can have many applications. Militarily, it provides the deployed land, air and sea vehicles with a complete view of the battle map and potential enemy targets. In space exploration, it presents the manned or unmanned vehicles with an exhaustive list of areas to be scanned and investigated. Nodes with a complete view of targets are faced with a dilemma of who should service what with respect to a specific metric (e.g. distance, energy). The decision on the course of action is very crucial. Multiple task assignment techniques have been proposed. Some require multiple inputs while others depend on one parameter. However no comparative study has been performed to list the pros and cons of selecting a technique.

1.3 Research Purpose

The purpose of this study is to build a communication network for randomly dispersed nodes. The algorithm combines adaptive resonance theory and Maxnet neural networks to cluster nodes. It uses integer linear programming for dominant sets to elect a cluster head. Gateways are selected using a willingness function that combines node degree, distance to cluster center,

available power and transmission energy. The resultant is a technique suitable for an on-demand or periodical topology configuration.

The second purpose of this study is to investigate the impact of the communication topology on the average state updates required in consensus building. Random topologies are generated. Few topology metrics, namely the algebraic connectivity ratio, average path, average cluster coefficient, average matching index, modularity and the average participation, were evaluated. The objective is to extract the most important metrics influencing consensus building.

Another goal of this study is to compare three techniques for task allocation. The comparison focuses on fairness, cost of assignment and workload distribution. The aim is to select a technique that provides a balance between these three characteristics. As part of the consensus building, a task assignment is required to determine the course of action for each node.

CHAPTER 2

LITERATURE SURVEY

2.1 Topology Creation

Different heuristics have been proposed in the literature to create clusters, elect cluster-heads and select gateway nodes. Some use a node identifier to elect cluster-heads. This approach, also known as lowest ID heuristics or identifier-based clustering, was introduced by Baker and Ephremides [1]. It assigns a unique ID to each node and elects the node with the lowest ID as a cluster-head. A cluster is then created by connecting all nodes with higher IDs to the cluster-head. A node is selected as a gateway, only if it lies within the transmission range between two cluster-heads.

Clustering based on node degree, max degree, or connectivity, is another commonly used approach. It was first introduced by Gerla [2]. A node with higher degree is more likely to be elected as a cluster-head. Using any protocol for neighbor discovery, a node with a maximum number of neighbors is elected to become cluster-head. If a tie occurs, the node with lowest ID is selected. The neighbors are then connected to the closest cluster-head to form a cluster. Only one cluster-head is allowed per cluster, and cluster-heads are connected to each other to act as gateways.

Another approach is to assign weight to a node based on its readiness and disposition to become a cluster-head [3]. A node is elected as a cluster-head if its weight is higher than its neighbors. There are different ways of assigning weights. Some techniques are based on node energy, position, degree, speed, direction, et cetera. Others use a probabilistic approach.

Tan et al. proposed a priority based adaptive topology management approach that uses a heuristic weight function based on the distances to neighbors, current communication state, energy level and node speed [4]. The node with higher degree is elected as a cluster-head.

Heizelman et al. proposed a low energy adaptive clustering hierarchy in which each node uses a probability value to elect itself as a cluster-head [5]. This approach is based on the number of cluster-heads and randomized rotation of cluster-heads.

2.2 Topology Impact on Consensus

Recent interest in consensus inference from topology structure has led to better understanding of parameters driving node dynamics in a network. Different approaches were proposed in the area of synchronization and consensus building. Some studies focus on discovering the relationship between the structural properties and the convergence; others tackle the problem from a numerical study point of view.

Comellas et al. characterized the influence of topology by finding the relationship between the algebraic connectivity, the convergence ratio in terms of structural properties [6]. The result is that convergence is sensitive to some structural properties.

Attay et al. on the other hand, compared the structural properties approach versus the spectral method [7]. Their conclusion is that topology properties fail, in some cases, to characterize the synchronization.

Furthermore, Jun-Zhong et al. focused on few commonly used structural properties [8] and reevaluated their effect on consensus. Their numerical simulations revealed that structural properties are inter-related and synchronization is difficult to predict even if the properties are separated.

In a different approach, Hovareshti et al. defined a new measure for the convergence and applied it to small world networks [9]. The aim is to propose a design guideline for building a network which maximizes the convergence.

In a similar way, Kar et al. conducted a series of numerical studies measuring the convergence speed on different type of topologies [10]. They reached the conclusion that expander graphs perform much better than structured graphs.

2.3 Task Assignment

Several solutions exist for task allocation in multi-robot systems. The technique presented by Sander et al. makes use of Delaunay triangulation to allow each node to identify a small set of adjacent nodes [11]. The objective is to move toward the target and away from other adjacent nodes.

Sujit et al. presented a negotiation scheme among a swarm of UAVs (Unmanned Ariel Vehicles) [12]. Equipped with target detection sensor, the UAVs locate a target and relay its coordinates to their closest neighbors. Negotiation among UAVs decides on who should attack the target. In another work, Sujit et al. also proposed team theory concepts to build a consensus among UAVs [13]. The objective is to minimize the cost and maximize the payoff. This approach requires no communication among UAVs.

Campbell et al. consider the case where robots are ignorant of their capabilities [14]. They proposed two learning techniques: greedy and best fit. Through a series of experiments, they showed that learning is important factor in task allocation.

Matarić et al. presented a technique based on bidding [15]. The focus is on two aspects: commitment and coordination. The robot has to decide whether to finish its current assignment or bid on other tasks and whether to base its action on local versus global information.

Butterfield et al. presented a probabilistic Markov random field model [16]. Two types of interactions are proposed and combined into a joint probability distribution that dictates the global decision. A local evidence function is defined to express the observation and the action variables of each robot. A pair-wise compatibility function is then defined to decide the action.

Goldberg et al. presented a market-based planning approach for task allocation [17]. They extended the three layer architecture (planning, executive and behavioral) to robot's decision making. At the planning layer, the trader component facilitates the auctioning and bidding of tasks. The scheduler component determines the feasibility and cost for the trader. At the executive level, the tasks are executed and synchronized. At the behavioral level, robots create feedback loops for control and coordination.

CHAPTER 3

TOPOLOGY CREATION

3.1 Introduction

Mobile sensor networks are characterized by frequent topology updates and self-configuration. Networks are created spontaneously whenever nodes are within the transmission range of each other. The arrival and departure of nodes in a network is an on-going dynamic process due to nodes high mobility. This dynamic nature causes the nodes in mobile sensor networks to lose connectivity. Strategies are required to keep the network connected and adaptable to frequent changes in the network.

Although suitable for many applications, the unpredictable overhead generated by frequent topological changes can be cumbersome. In order to handle this problem, some strategies tend to focus on node power control mechanisms [18, 19, 20, 21], while others emphasize node hierarchy [22, 23, 24, 25]. Power control schemes tune power at each node to ensure the closest neighbor connectivity and thus the overall network connectivity. The hierarchical based approach, also known as clustering, consists of dividing nodes into two subsets. A set of cluster-heads and a set of nodes associated with cluster-heads. Connection among cluster-heads is established through selected gateway nodes. Regardless of the approach, the challenge is to keep the network stable as long as possible before a re-organization becomes necessary.

In this chapter, we propose a periodic and on-demand topology reconfiguration framework for a mobile sensor network that results in enhanced connectivity and performance. The re-configuration is on-demand when nodes leave or join the network and periodic when

employed as an alternative approach to topology control and maintenance through scheduled re-organization.

The proposed approach consists of four stages. The first stage utilizes a combination of adaptive resonance theory (ART) and Maxnet to cluster nodes. Maxnet's weight update equation helps in finding adjacent nodes in the network. The ART approach contributes to the formation of new clusters when adjacency conditions are not satisfied. The second stage addresses the election of a cluster-head for each cluster to facilitate intra-cluster communication. The selection is performed using an integer linear programming formulation on the adjacency or neighborhood matrix. The third stage deals with the establishment of inter-cluster communication. Gateway nodes are selected based on a willingness function that combines node degree, distance to cluster center, available power and transmission energy. The fourth stage consists of establishing an optimal connection between clusters. The task is also performed using an integer linear programming formulation.

The chapter is organized as follows. Section 3.2 presents the mathematical concepts used in the proposed model. Section 3.3 outlines the proposed on demand communication topology update strategy. In this section, the neural network approach for clustering nodes based on range and proximity is described. It describes the selection of cluster-heads using the concept of minimal dominant sets. It describes the willingness function used to select the gateway nodes required for communication among clusters. It describes the approach for cluster inter-connection. Section 3.4 analyzes the complexity of the method.

3.2 Assumptions and Mathematical Preliminaries

In this section, we present the assumptions and the mathematical concepts used in the proposed model. First, nodes are to be confined to a geographic area. Nodes outside the defined perimeter belong to a different area. Communication between different areas is not considered in this paper. This assumption is required to allow nodes to discover each other via broadcast messages or through neighbor discovery, which is not a trivial task. Ideally, an omni-directional antenna with variable transmission range to cover the confined area would be appropriate. This requirement is necessary since nodes have to adjust their transmission range for inter-cluster communication. An alternative solution is a directional antenna, which is likely to detect more neighbors because of longer transmission range but requires processing antenna direction. For a distributed processing, nodes are assumed to determine their respective location and include it in their advertisements and communication along with their transmission range. Nodes should share any knowledge about other nodes' position and transmission range.

The second assumption requires that nodes communicate through their elected cluster-head. Communication among peers is not allowed. Instead, communication between cluster-heads is performed via a designated gateway. A backup gateway, if available, will act as a normal node to prevent loops.

In idle state, nodes have a limited transmission range. The proposed approach uses this limitation during cluster formation and assumes that the nodes will adjust their transmission range afterwards.

We represent the network of nodes as an undirected graph $G=(V, E)$ where V is the set of vertices and E is set of edges. The neighborhood $N(v)$ of a vertex v consists of a set of vertices

adjacent to v, that is, $N(v) = \{u \in V: uv \in E\}$. Thus, the adjacency matrix A and neighborhood matrix N can be represented as follows:

$$A = [a_{u,v}] = \begin{cases} 1 \; if \; uv \in E \\ 0 \; otherwise \end{cases}$$

$$N = A + I_n : \; I_n \text{ is the identity matrix.} \tag{1}$$

We also use the concept of domination in graph theory. A node u dominates another node v, if there is a link between u and v. There are several ways to define a dominant set in a graph, each of which illustrates a different aspect of dominance [26, 27]. The definitions we followed are described below and they will be used in section 3.3. We refer the reader to

Definition 1 [28]: *A set S of vertices in a graph G = (V, E) is a dominant set if and only if all the following conditions hold true.*

(a) $\forall \; v \in V\text{-}S, \; \exists \; u \in S$ *such that v is adjacent to u.*

(b) $\forall \, v \in V\text{-}S, \; d(v, S) \leq 1$*, that is, no more than one edge is permitted between v and one vertex in S.*

(c) $\forall \; v \in V\text{-}S, \; |N(v) \cap S| \; \geq 1$*, that is, every vertex* $v \in V\text{-}S$ *is adjacent to at least one vertex in S.*

In our approach, we are interested in finding a *minimal dominating set (MDS) S* such that no proper subset $S' \subset S$ is a dominant set and $\bigcup_{v \in S} N(v) = V(G)$.

Theorem 1 [28]: *A dominating set is a minimal dominating set if and only if for each vertex* $u \in S$*, one of the following two conditions holds:*

(a) u is an isolated vertex of S.

(b) $\exists \; v \; \in V\text{-}S$ *for which* $N(v) \cap S = \{u\}$.

3.3 Proposed Model

This section describes the steps involved in building a communication topology for a group of nodes that are randomly placed. In the first step, nodes within each other's range are grouped into clusters. In the second step, cluster-heads are selected to facilitate intra-cluster communication. In the third step, gateway nodes are selected to facilitate inter-cluster communication. In the fourth step, inter-cluster links are established.

3.3.1 Clustering

Clustering is the process of partitioning a set of nodes into subsets that share a common characteristic. The most common criterion to group nodes is proximity. There are two methods to learn how nodes are organized: supervised and unsupervised. We focus only on unsupervised learning and particularly on competitive neural networks (NNs). We combine the concepts of two NN models Maxnet and ART to develop a clustering strategy.

Maxnet is an NN that does not require training, serves as a classifier and is used in identifying the winning node with the highest weight [29]. In our implementation, we are interested in finding nodes that satisfy the constraint of being within each other's range. Our approach starts with assigning initial weights to links based on the distance between nodes. It uses an adjacency function $f(x)$, as an indicator of an adjacent node; a weight update function to pick the "winning" node in the competition and a stopping condition to halt the weight update [29]. The adjacency function and the weight update function also used in evaluating the adjacency *A* and neighborhood *N* matrices. The adjacency function is described by

$$f(x) = \begin{cases} x & if\ x > 0 \\ 0 & otherwise. \end{cases} \tag{2}$$

The weight update function is described by $w_{ij}(new) = f\left(w_{ij}(old) - \beta \sum_{k \neq j} w_{ik}(old) \right)$, (3)

where $0 < \beta < \frac{T_i}{2}$, w_{ij} is the distance between node *i* and *j*, and T_i is the transmission range. The iterative process of the weight update given in equation 3 reduces the distance between any given two nodes (say, *i* and *j*) by a small value ($\beta \sum_{k \neq j} w_{ik}(old)$) and tests whether the weight reduces to zero. Note that *f*(*x*) also serves as a stopping condition. The weight update function requires that the quantity subtracted satisfies the inequality $\sum \left(\beta \sum_{k \neq j} w_{ik}(old) \right) \leq T_i + T_k$ A value of zero in any row of the weight matrix (w_{ij}) indicates that a within-range adjacent node has been identified. This concept is illustrated in table 1. The value of zero between (N2,N3) and (N3,N4) indicates an adjacency between these nodes which is illustrated in Figure1

The second concept borrowed from ART network allows the user to control the degree of similarity of patterns placed in the same cluster, and in turn, to control the number of clusters [29]. New inputs are first tested against the existing clusters. If no cluster fits the new input, a new cluster is created. Inputs with similar patterns are grouped in the same cluster. The creation of new clusters relies on a vigilance condition defined as follows.

$$Dist(i, j) \leq T_i + T_j \quad (4)$$

The algorithm creates a cluster for each pattern that does not satisfy the vigilance condition. Note that the right hand side of the vigilance condition can be controlled by the user.

TABLE 1

FINAL WEIGHTS COMPUTED BY THE WEIGHT UPDATE FUNCTION

	N1	N2	N3	N4
N1	1	0.32	0.22	0.04
N2	0.32	1	0	0.01
N3	0.22	0	1	0
N4	0.04	0.01	0	1

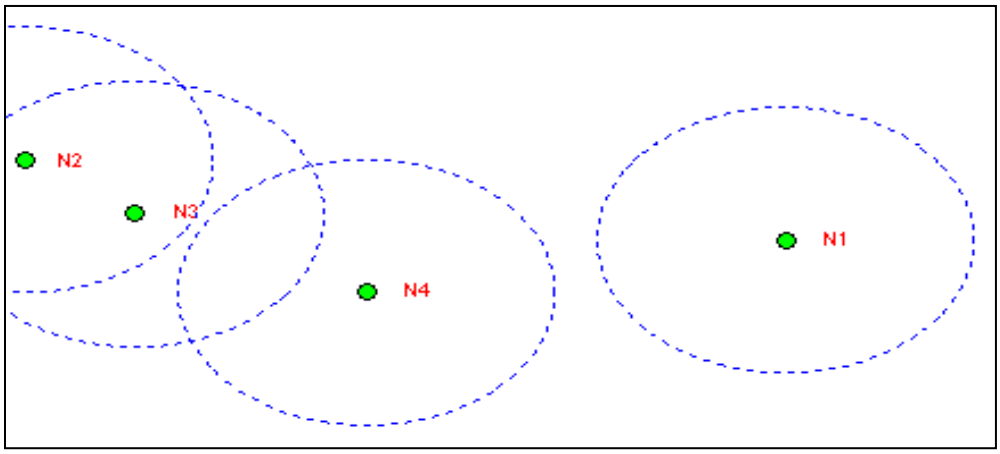

Figure 1*:* The resultant clusters related to table 1. The dotted circles indicate the transmission range of each node.

3.3.2 Computing Minimum Dominant Set

After evaluating the adjacency matrix and determining the clusters in step 1, the next step is to elect the cluster-head set S using the concept of minimum dominant set (MDS). Computing MDS has been an active area of research in many fields. Among the earliest formulation of the problem was the placement of the five queens in a chessboard game. The queens have to be in a position either to occupy or attack any square. Many heuristics, mostly in graph theory and ad hoc networks, have been proposed to solve the problem of defining the MDS. Whether centralized [30, 31] or distributed [32, 33], the proposed algorithms try to reduce the problem

complexity (which is NP-hard). In our approach, we use a technique that is pseudo-polynomial, since it uses integer linear programming (ILP), which is known to be NP-complete [34].

In the proposed approach, MDS is computed using a characteristic function:

$$f : V(G) \longrightarrow \{0,1\} \tag{5}$$

, which satisfies

$$f(v) = \begin{cases} 1 & if\ v \in S \\ 0 & otherwise \end{cases} . \tag{6}$$

The characteristic function states that v is a cluster-head if it belongs to S. This function will be used to assign values to the variables in the linear programming formulation.

There are two ways of defining the MDS [28, 35], both of which yield the same results during simulation. The first approach uses the adjacency matrix, in which a vertex v is considered to dominate the vertices in its neighborhood but not itself. It is equivalent to finding the vertices that have maximum neighbors and cover the whole graph:

$$AX \geq \overline{1}_n, \tag{7}$$

where A is the adjacency matrix, $X = (x_1, x_2, ..., x_n)$ is the column of variables to be evaluated, $\overline{1}_n$ denotes the column n-vector of all 1's and n is the number of nodes.

The integer programming formulation for minimum cardinality domination set γ(G) is described as:

$$\gamma(\mathrm{G}) = \min \sum_{i=1}^{n} x_i \tag{8}$$

$$subject\ to\ \ AX \geq \overline{1}_n$$

$$with \qquad x_i \in \{0, 1\} ,$$

which translates to minimizing the number of nodes that have the highest number of adjacent nodes.

The second formulation which uses the neighborhood matrix, assumes that each vertex can be dominated at most once and seeks to achieve as much domination as possible. Efficient domination happens when the maximum number of dominant vertices can be found. The integer programming formulation for computing the efficient dominating set *F(G)* is given by:

$$F(G) = \max \sum_{i=1}^{n} (1 + \deg(v_i)) x_i \tag{9}$$

$$\textit{subject to} \quad NX \leq \overline{1}_n$$

$$\textit{with} \qquad x_i \in \{0, 1\},$$

where N is the neighborhood matrix and $\deg(v_i)$ is degree of node v_i. This formulation translates to maximizing the number of neighbors of v_i.

At this stage, computing MDS by either method yields a set of all possible dominating nodes for that graph. As a result, some clusters end up with more than one dominant node. Selecting a cluster-head within one cluster requires the removal of the other candidates. The process of eliminating dominant nodes within one cluster is based on node degree. The highest degree node is selected as cluster-head. If a tie occurs, the node with the lowest *id* is selected.

3.3.3 Selecting Gateways

In the previous sections, we discussed ways to create clusters and to elect cluster-heads within each cluster. In this section, we discuss the process of gateway selection. The purpose of the gateway selection process is to establish and facilitate communication between clusters. The mechanism that drives the selection of a gateway among multiple nodes is a process that

involves two steps: a selection step and a decision step. We introduce a new "willingness function" in the selection process. The decision process to elect a gateway relies on the node with the highest willingness value. The pool of nodes is limited to the nodes within the same cluster excluding the cluster-head. The proposed willingness function incorporates four factors described below. The proposal was inspired by [36, 37] for electing a cluster-head. We adopted this technique for gateway selection by incorporating the following parameters.

3.3.3.1 Distance to the Centroid

This parameter denotes the distance of a node from the cluster's center which is defined as follows:

$$C\left(x_{Centroid}, y_{Centroid}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(Rx_i, Ry_i\right), \tag{10}$$

where n is the number of nodes in the cluster. Rx_i *and* Ry_i are the x and y coordinates of node i. The distance from a node j to this centroid is described by

$$DistToCent_j = dist\left(j, C\right), \tag{11}$$

where j is any node under consideration for gateway selection. A node closer to the centroid is a lesser candidate to act as a gateway and a peripheral node is more likely to be selected as a gateway.

3.3.3.2 Transmission Energy

This parameter measures the transmission energy required by each node to reach all other nodes in the same cluster. This parameter is given by:

$$TrE_j = \exp\left[T_j \sum_{i=1}^{n} d_{ij}^2 / \left(1 + \sum_{i,j=1}^{n} d_{ij}^2 \right) \right] - 1, \quad (12)$$

where T_j is the transmission range and d_{ij} is the distance between node i and node j. A node that is farther from all other nodes requires more transmission energy to communicate with other nodes. Unlike the *DistToCent* parameter, *TrE* is directly linked to the required battery power.

3.3.3.3 Node Degree

This parameter reflects the node's neighbors that are within its transmission range.

$$Deg_j = \sum_{i \neq j} (dist(i,j) < T_j), \quad (13)$$

where T_j denotes the transmission range of node j.

3.3.3.4 Available Energy

This parameter is a gauge for a node's available energy. There are three types of energy costs associated with this parameter. As time progresses, there is a cost for sitting idle and listening for traffic (IdleE), a cost for movement (MvE) and a cost for transmitting (TrE). MvE and IdleE are mutually exclusive.

$$AvailE_j = AvailE_j - IdleE_j - TrE_j - MvE_j \quad (14)$$

(MvE > TrE > IdleE)

The willingness function (WF) is a combination of these parameters and is defined as follows:

$$WF = 1 - \alpha * DistToCen - \beta * TrE - \lambda * Deg - \gamma * AvailE \quad (15)$$

The values of α, β, λ and γ are user defined parameters and add up to 1. As an example, the values assigned for the initial topology creation are α=0.4, β=0.1, λ=0.5 and γ=0, since we assume that all nodes have equal energy to begin with. For a given cluster, the willingness function is evaluated for all nodes of that cluster. The node, excluding the cluster-head (CH3), with the highest willingness value is selected as the gateway for that cluster.

Table 2 and Figure 2 illustrate the process of gateway selection. Table 2 shows that node 2 which has the highest willingness value has been selected as gateway (G2). Figure 2 shows node positions, location of the centroid and the cluster head.

TABLE 2

EVALUATION OF THE WILLINGNESS FUNCTION FOR FOUR MOBILE NODES

Node	LocToCent	TrE	Deg	Willingness
1	0.3103	1.045	1	0.2714
2	**0.2797**	**0.866**	**1**	**0.3015**
3	0.0647	0.215	2	-0.0474
4	0.0945	0.223	2	-0.0601

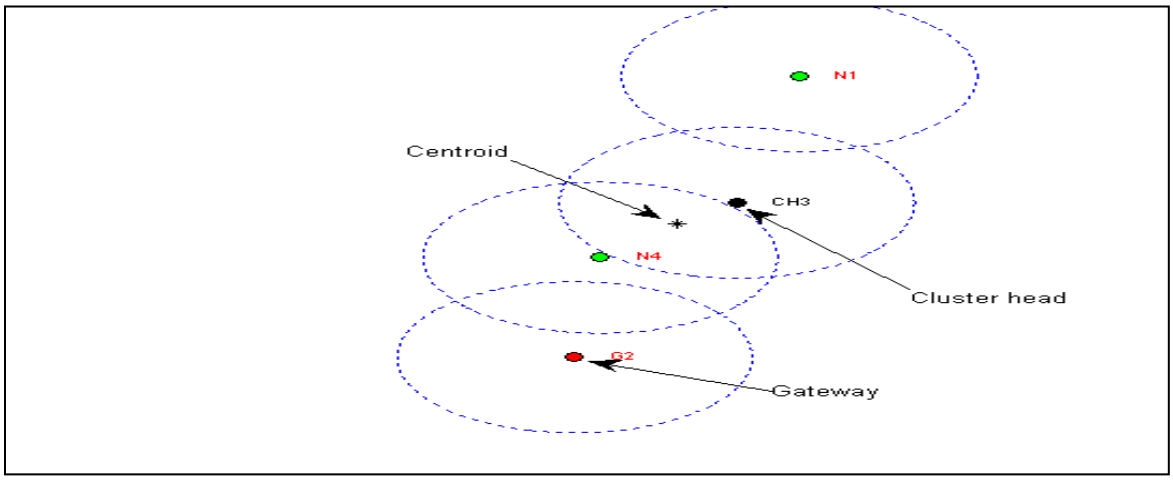

Figure 2: Node positions corresponding to Table 1.

3.3.4 Cluster-head and External Gateway Link

Cluster-heads and gateways are the fundamental components in a mobile sensor communication network. All communications need to pass through them before reaching the rest of the nodes. Once the cluster-heads are determined, the challenge is to find a link that connects them under the constraint that the communication between cluster-heads is via gateways. However, due to the randomness of node placement, this constraint is not always guaranteed to be satisfied.

The purpose of establishing a cluster-head and an external gateway link is to connect a gateway with an external cluster-head (ECH) for a complete topology. There are different scenarios that need to be considered. Generally, the cluster-head is not a candidate to be elected as a gateway. But there may be cases where a cluster is formed by one single node that acts as a cluster-head and gateway at the same time. To solve this problem, two separate approaches are proposed. The first one tackles the normal case, and the second approach deals with the exceptional case.

The first approach, in which one gateway is linked to one external cluster-head, can be compared to a perfect matching graph assignment with equal set cardinality. Thus, it can be solved by this simple expression:

$$\begin{aligned} &\min \sum_{i=1}^{n}\sum_{j=1}^{m} c_{i,j}x_{i,j} \\ &\textit{subject to} \sum_{i=1}^{n} x_{i,j} = 1 \\ &\qquad\qquad \sum_{j=1}^{m} x_{i,j} = 1 \\ &\textit{with} \qquad x_{i,j} = \{0,1\}, \end{aligned} \tag{16}$$

where $x_{i,j}$ is the variable to be evaluated and $c_{i,j}$ is an element of the distance matrix between all gateways and external cluster-heads belonging to multi-node clusters.

The second approach requires that a gateway may connect to more than one external cluster-head belonging to a single-node cluster; and that each external cluster-head is linked to one external gateway. The formulation of this problem can be represented as a minimum weight b-perfect matching problem and solved using the following integer programming expression.

$$\begin{aligned} &\min \sum_{i=1}^{n}\sum_{j=1}^{m} c_{i,j}x_{i,j} \\ &\textit{subject to} \sum_{i=1}^{n} x_{i,j} \geq b_i \\ &\qquad\qquad \sum_{j=1}^{m} x_{i,j} = 1 \\ &\textit{with} \qquad x_{i,j} = \{0,1\}, \end{aligned} \tag{17}$$

where $c_{i,j}$ is an element of the distance matrix between all gateways and external cluster-heads belonging to single-node clusters and b_i is the minimum number of external links allowed per gateway ($b_i = 1$ in our case).

3.4 Complexity

Analyzing the efficiency of an algorithm is of primary concern to any user. The goal is to have an idea on how much resources (time, storage) are required for the technique to execute. Run-time analysis objective is to estimate the increase in time when the number of inputs increases and determine an upper bound or limit for the execution time. Analyzing a technique necessitates analyzing each of its components.

The re-configuration method is composed of three parts: clustering, gateway selection and inter-cluster link establishment. Clustering involves three nested loops $\theta(n^3)$ and solving an

ILP problem. Gateway selection is accomplished by two nested loops $\theta(n^2)$. Inter-cluster link establishment is computed using two ILP formulations. ILP uses a branch and bound based search technique [38, 39], which in turn, uses a tree structure to solve the problem. It partitions the feasible region into subsequent smaller subsets and then calculates bounds for each subset. These bounds are then used to discard some subsets and to update the current solution by a better solution. The iterative process stops when no feasible solution or better solution could be produced. In general for n ≥3, the optimal solution will result in $(n-1)/2$ decision variables x_i equal to 1 and $(n+1)/2$ decision variables x_i equal to 0, which suggests that ILP is at least $\theta(2^{(n+1)/2})$ exponential. The approach has an exponential order. It is suitable for applications that require a small number of nodes.

CHAPTER 4

TOPOLOGY IMPACT ON CONSENSUS

4.1 Introduction

In many fields, multi-agent systems are often characterized by the interactions that take place among the agents. In biology for example, the collective behavior of cells and tissues is a direct consequence of the properties of their molecular constituents. In computer networks, the overall performance is measured by each station's throughput, security, scalability of protocol, et cetera. In manufacturing, the quality of a product is evaluated by the functionality, reliability and maintainability of each component. The performance of such a complex system is affected by the connectivity and placement (topology) of its constituents.

Complex systems of collaborative multi-agent systems have to address coordination problem as well. Constructing a communication topology for randomly dispersed nodes is necessary for exchanging information. In a mobile network, the initial scope of information available to each node is usually limited by the information collected locally. Increasing this scope of information requires either collecting more data or requesting it from other nodes. The iterative process of exchanging local information among nodes followed by inferencing in order to gain global perspective of a problem is referred to consensus building. There are many factors that influence the consensus building process, communication topology being the most important of all. The relationship between the communication topology and the consensus process, at present, seems to be a black box. Studies aimed at evaluating the impact of the characteristics of a communication topology on the process of consensus building do not exist in the literature as of now.

In order to investigate the intricate connectivity and coordination, the system has to be translated into an abstract concept that can be mathematically modeled. The model is then subjected to a systematic characterization which seeks to uncover its characteristics. These characteristics in turn, provide insights into the design and organization of the network that affect the consensus building process. Statistical techniques such as regression analysis and factor analysis can then be applied to show the interactions and interdependencies among the topological characteristics, and the impact of these structural properties on consensus building. The most significant properties can be included in the model for consensus building and the performance of such a model can be analyzed in detail.

4.2 Topological metrics

We begin with a set of topological metrics that we want to evaluate [10, 40, 41, 42, 43, 44, 45, 46, 47]. Several structural properties and metrics were defined in the literature for networks based on intuitive concepts [48, 49, 50, 51]. Due to the difficulty in enumerating all properties and metrics, it becomes a necessity to choose a suitable subset. The challenging question then is to identify the most relevant properties and metrics? In our approach, we focused on introducing metrics that were not investigated extensively in the literature while keeping some of the well investigated metrics (average path and cluster coefficient). Then, we establish a relationship between the new set parameters and other parameters that are commonly used in the existing models in the literature.

4.2.1 Average Path

In graph theory, the minimum distance d_{ij} between two vertices i and j is measured by the length of the shortest path between the vertices or the minimal number of edges that can be

traversed. The average path length $d = \langle d_{ij} \rangle$ is defined as the average distance between all pairs of vertices. Computing the shortest path between two vertices is not straightforward. The commonly used algorithm is Floyd-Warshall's, which runs in $\theta(n^3)$ and returns shortest path between all pairs of vertices [52].

4.2.2 Average Cluster Coefficient

Cluster coefficient is another characteristic that relates to the internal structure of a network, more precisely its cohesiveness. It measures the probability that two vertices with a common neighbor are connected. Given a vertex i with n_i neighbors, $\frac{n_i(n_i-1)}{2}$ possible edges could exist between the neighbors. The clustering coefficient C_i could be defined as the ratio of the actual number of edges E_i over all the possible edges.

$$C_i = \frac{2E_i}{n_i(n_i-1)}. \tag{18}$$

The average cluster coefficient is defined as $C = \frac{1}{n}\sum_{i=1}^{n} C_i$, where n is the number of nodes.

4.2.3 Average Matching Index

Matching index is a basic measure for neighborhood analysis. It quantifies the topological overlap between two nodes by measuring their similarity in terms of common shared neighbors. The similarity is given by the following expression:

$$M_{ij} = \frac{\sum common\ neighbors}{\sum total\ \ number\ of\ \ neighbors} = \frac{\sum_{k,l}^{n} a_{ik}a_{jl}}{n_i + n_j - \sum_{k,l}^{n} a_{ik}a_{jl}} \quad , \tag{19}$$

where a_{ij}, a_{jl} are elements of the adjacency matrix and n_i, n_j are the respective number of neighbors of vertices *i* and *j*. The average matching index is defined as:

$$AMI = \langle M_{ij} \rangle = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} \,. \tag{20}$$

4.2.4 Average Participation Coefficient

Participation coefficient is characteristic that defines the role of each node in a complex network. The role fulfilled by a vertex is an important factor since it decides the node's connections. A cluster-head or a gateway has more connections than a peripheral node since they act as facilitators in addition to their default role. This leads to the interpretation that the participation coefficient provides information about the link distribution among different clusters. A higher value indicates that the network has a higher ratio of nodes to clusters and vice versa. Computing the participation coefficient P_i of vertex i is given by:

$$P_i = 1 - \sum_{c=1}^{n} \left(\frac{k_{ic}}{k_i} \right)^2 , \tag{21}$$

where k_{ic} denotes the degree of i within its cluster c and k_i is the total degree of i . The average participation coefficient is defined as: $P = \langle P_i \rangle$.

4.2.5 Algebraic Connectivity Ratio

Another commonly used property in measuring the connectivity in a graph is the algebraic connectivity parameter. It is an important factor in consensus building since it directly affects the network synchronization. This parameter also provides information about the robustness of a topology in terms of link and node failure. The higher the value the more failure

tolerant is the network. In a given topology, this parameter is evaluated by the second-smallest eigenvalue of the Laplacian matrix L. The ratio γ of the second-smallest eigenvalue over the last eigenvalue defines the algebraic connectivity ratio.

$$\gamma = \frac{\lambda_2(L)}{\lambda_n(L)} \tag{22}$$

The eigenvalues of L are ordered such that they satisfy: $0 = \lambda_1(L) < \lambda_2(L) < ... < \lambda_n(L)$.

4.2.6 Modularity

Cluster coefficient, as defined previously, measures the bonding between nodes. Modularity, on the other hand, quantifies the strength of clusters and evaluates the quality of the clustering. When "good" clustering occurs, intra-cluster connections are dense and inter-cluster links are sparse, whereas, when "bad" clustering occurs, nodes form a cluster on their own. One of the definitions of modularity is the difference between the fraction of intra-cluster edges over the total network edges and the fraction of extremities that connect two clusters and is given by:

$$Q = \sum_{i=1}^{n} \left(e_{ii} - b_i^2 \right), \tag{23}$$

where, e_{ij} represents the fraction of within-cluster edges, $b_i = e_{ii} + \frac{1}{2}\sum_{i \neq j}^{n} e_{ij}$ and e_{ij} represent the fraction of edges having one extremity in cluster i and the other in cluster j. Q is proportional to quality and tends to be negative for "bad" clustering.

4.3 Consensus Algorithm

In the previous sections, we presented the steps involved in building our communication network and provided a brief description of some structural properties we intend to extract from

the communication topology. In this section, we present a description of the distributed implementation of the consensus building algorithm along with the mathematical formulation of the information flow that will be used as the main metric to evaluate the average state update.

Before presenting the consensus building algorithm, we need to define initial target awareness for each robot. At this level, robots will have partial or no information about the targets location. The objective of each robot is to build the list of all targets through the consensus algorithm. Since target discovery is not within the scope of this research, we experimented with two simple approaches. The first one assigns randomly a target to a robot. The advantage of this method is its fairness in target awareness. However it fails on proximity criteria (target assigned to the closest robot). The second approach uses a self-organizing map principle when the winner (closest) takes all. It has the advantage of being close to real-world scenario. However, its shortcoming is the lack of fairness. Many robots end up with no initial information about targets. To remedy this issue, targets were randomly positioned to increase the chance of partial knowledge. Having built a target scope for each robot, the next step is to exchange these initial scopes in order to build a consensus (final list of targets). The task is performed through an iterative process that is divided into two data flows processes: bottom up and top down (figure 3). Bottom up is used by a child to update its parent node. The top down flow is utilized by a parent to update its child. However, due to randomness in the placement of the robots, there are cases where nodes form clusters on their own. Figure 4 illustrates an example of isolated cluster-heads (ICHs) where CH3 is the root and CH2 is the last descendent. To accommodate this case and reach a consensus among ICHs, the algorithm includes communication data between ICHs. Thus,

Bottom up flow:

- Each ICH that has a parent ICH sends its target scope to it.
- Each ICH or Cluster head (CH) that has a parent Gateway sends its target scope to it.
- Each gateway or regular node sends its target scope to the parent cluster head.
- During this bottom up data forwarding, each node receiving a target scope updates its scope.

Top Down flow

- Each CH that has a child gateway or regular node sends its target scope to it.
- Each gateway that has a child ICH or CH sends its target scope to it.
- Each ICH that has a child ICH sends its target scope to it.
- Similarly, each node receiving data, updates its scope.

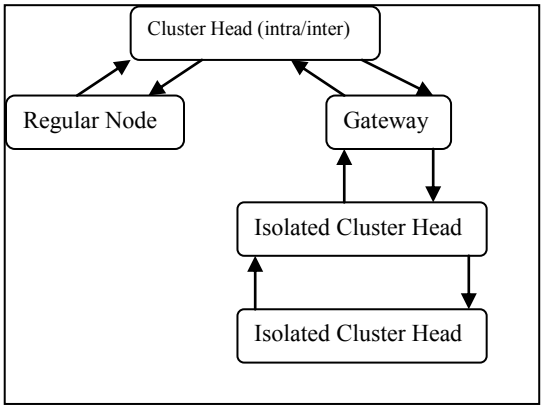

Figure 3: Diagram showing the flow of information between different types of nodes.

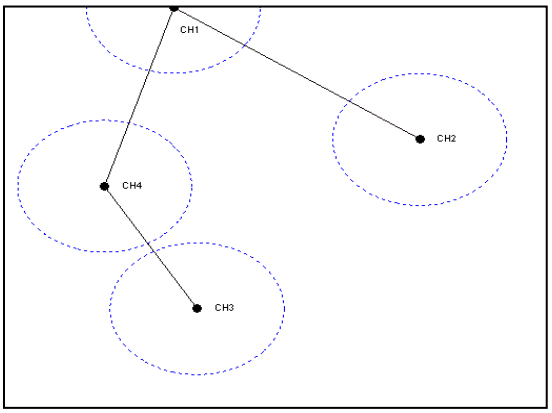

Figure 4: Nodes forming clusters on their own where CH3 is the root.

In summary, before the "target discovery" process, all the nodes in the network are in a state of zero knowledge $S_i(t=0)=0$, where $S_i(t)$ is the target scope of robot *i* at time *t*. After the initial discovery of targets, the state of some robots changes to reflect the partial awareness $0 \le S_i(t=1) \le n$ assuming that there are *n* targets to discover. At the end of the iterative

communication process and exchange of information, the robots reach the saturation state, where they all have a complete and identical view of the targets list.

In general, the state of a robot i changes whenever it receives a state update from either its parent or descendant(s). This can be formulated by the following equation.

$$S_i(t) = S_i(t-1) + \sum_{j=1, j\neq i}^{m} a_{ij} s_j(t), \tag{24}$$

where m is the number of nodes, a_{ij} denotes the adjacency of node i in the network topology and emphasizes that only the parent or descendent(s) are allowed to share their state s_j with it.

However, not all exchange of information between a node i and its parent or descendant(s) results in updates of S_i. Sometimes, it takes a few iterations for new information to travel from one extremity to the network to the other, meanwhile, nodes in the middle exchange the same state. To incorporate this into our equation, we define an activation function

$$\Phi(x) = \begin{cases} x & \textit{If } x \textit{ is a new target.} \\ 0 & \textit{Otherwise.} \end{cases}$$

Equation 24 then becomes:

$$S_i(t) = S_i(t-1) + \sum_{j=1}^{m} \Phi\left(a_{ij} S_j(t)\right) \tag{25}$$

At any given time, a robot must check whether it reached the saturation state or not. To accomplish that, it evaluates its state through the following saturation function:

$$\sigma(s_i) = \frac{2}{1 + e^{\left(\frac{n - S_i}{an}\right)}}, \tag{26}$$

where a is a constant. If the returned value is 1, the robot's target list is complete. The iterative update process ends when all the robots reach their saturation.

To measure the consensus building, we define the average state changes as the total number of necessary updates performed by each robot to reach saturation over the number of robots as follows:

$$ASU = \frac{\sum_{i=1}^{m}\sum_{t=0}^{T} S_{i,t}}{m}, \tag{27}$$

where m is the number of robots, $S_{i,t}$ is the state update of robot i at time t and T is the saturation time.

Practically, in small world or random networks, a robot can verify whether it reached the saturation state or not without knowing in advance the number of targets to discover. It can be accomplished by evaluating the network size through its diameter, which increases logarithmically with the number of nodes [53]. Thus if the number of unchanged messages coming from one direction exceed *log(m)* (m number of robots), it is an indication that saturation had been reached.

4.4 Methodology

In any scientific field, studying the nature of parameters is necessary, but not sufficient. The study should also account for the relationship among the variables. The idea of variables being interrelated is a well-accepted concept that can be mathematically represented. It helps in better understanding the contribution of each parameter, by itself and through interaction, to the overall result. Hence the evaluation of parameter is an important factor in deciding whether to account for it or ignore it in the study.

Among the commonly used multivariable statistical methods that help scientists in analyzing the relationship between variables is factor analysis.

The aim of factor analysis [54, 55] is to reduce the set of observable variables to few hypothetical variables called factors. These factors are supposed to contain the essential information in the original variable set and are constructed in a way that reduces the overall complexity of the data by taking advantage of inherent interdependencies.

In addition to studying the interrelationships between the variables, the objective of this section is to partition data into mutually exclusive sets with different weights. The importance assigned to each variable within a set will also help in reducing the dimensionality of data.

The factor analysis method involves the following major steps:

- Collecting data and computing the correlation matrix.
- Extracting the unrotated factors.
- Rotating and interpreting the factors.

4.4.1 Data Collection and Correlation Matrix

The selection of the variables to investigate has a critical bearing on the output of the study. The problem becomes more complex when the conducted investigation is an exploratory one. The lack of knowledge about the underlying nature of variables can lead to the discovery of higher orders of overlapping or similarities among the variables that requires a reassessment of the variables under consideration.

As mentioned previously in section 4.2, the objective is to establish a bridge between newly introduced variables and some already studied parameters. The topological variables we intend to investigate using factor analysis are the following:

- Average path.
- Average cluster coefficient,

- Average matching index,
- Average participation coefficient,
- Algebraic connectivity ratio, and
- Modularity.

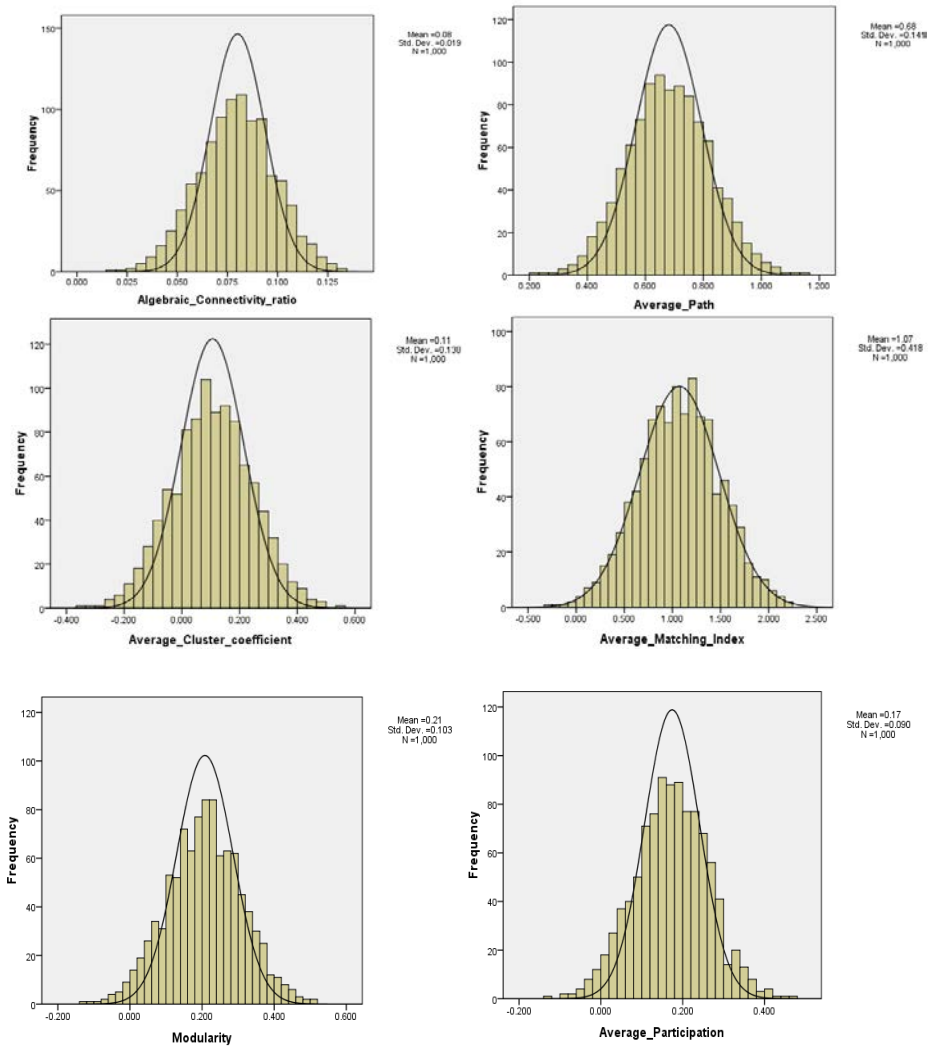

Figure 5: Histograms and normal distribution plots for the six variables using a sample size of 1000.

To perform a study using factor analysis method, there are some constraints that need to be satisfied. The first constraint deals with the measurement characteristics of the variables.

Ideally, factor analysis requires the data to be continuous and normally distributed (bell-shape). This requirement is important because it has a direct effect on the reliability of the correlation coefficient which is an indicator of the degree of correlation between variables. In practice, a bell shape is seldom satisfied. Prior to analysis, a pre-processing stage is sometimes required to clean up data from noisy, missing and outlier values. Since it is an exploratory area for us, we limited our pre-processing stage to normalizing data to fit a bell shape distribution. The initial test of normality on raw data revealed that none of the variables followed a normal distribution. The best goodness-of-fit value obtained was for a 4-parameters beta distribution. The histograms and distribution fitting of the six normalized variables are presented in Figure 5.

The second constraint deals with size of the sample. As the size increases the more stability is introduced in the evaluation of the correlation coefficients. Factor analysis recommends a sample size between 500 and 1000. Below that range, spurious variance distortion increases, and above that range no major gains are obtained.

After the data pre-processing step, the next crucial point in factorial analysis is the evaluation of correlation coefficients. This step helps in determining the relationship strength between the set of measures upon which the final decision will rely on. A major concern at this stage is the accuracy of the correlation coefficients. A result is statistically significant if it is unlikely to have occurred by chance. Usually a statistical test of significance is devised for each coefficient. It computes the probability p of a result being larger and compares it to the confidence level α. The result is said to statistically significant if $p \leq \alpha$. However, a statistical significant result is not always of practical significance. Since the conducted study is an exploratory one, the only option available to us is to rely on the statistically significant results. To compute the correlation matrix for the six topological metrics, we used Pearson's correlation

coefficients. Pearson's technique requires the data to be normally distributed and uses the covariance of the two variables over their standard deviation product to compute the elements of the correlation matrix [56]. The coefficients can take on values between -1 and +1. The result is a correlation matrix for each pair of variables along with a p-value for the test of significance for each coefficient. To interpret the correlation coefficients, statisticians recommend squaring the coefficients. The new coefficient is called a coefficient of determination. It provides information about the contribution of each variable to the total variance in one variable. For example, if the correlation between V_1 and V_2 is 0.5 then 25% of the variance in V_1 can be explained by the linear relationship between V_1 and V_2. The other 75% remains unexplained.

4.4.2 Factor Extraction

The general goal of factor analysis is to define and extract clusters of highly interrelated variables from the correlation matrix [57]. These clusters are called factors and can be written as a linear combination of the variables. Different techniques for factor extraction have been proposed. Each has its possibilities, limitations and advocates [58]. The techniques include principal component, maximum likelihood, alpha, image, centroid, minimum residual, et cetera. The most widely used is principal component. It has the characteristic of being suitable for exploratory factor analysis. However, some assessments and evaluations of these methods suggest that the best method to use when data is normally distributed is maximum likelihood because it evaluates a wide range of goodness of fit indexes for the model [56].

Choosing an extraction technique is only the first step in factor extraction. The next important step is to decide on the number of factors to retain. The main objective is not to over-extract or under-extract. But there is a general consensus among researchers that it is better to

extract too many factors than too few. At this stage, different criteria are proposed to stop the extraction process [53]. The most commonly used are the eigenvalues which are the sum of the factors squared weights. The cutoff point is a value above one. Another criterion based on discontinuity is the scree test [56]. It involves locating the point where the sudden drop of eigenvalues occurs on a scree plot. The points above it are generally the factors to retain. A third criterion relies on the desired percentage of variation to be explained by the factors. Extraction in this case, is subjective and relies on the user requirements on when to stop extracting.

In our investigation, we would like to experiment with principal components and maximum likelihood. The objective is to compare their results in order to choose the best method to use for our analysis. Maximum likelihood has the characteristic of providing the correct number of factors to extract due mainly due to the definition of the degree of freedom for the chi-square test which depends on the set of variables $df = \frac{(z-k)^2 - (z+k)}{2}$, where z denotes the number of variables and k is the number of factors to extract. Knowing the number of factors we will then use the method that is more accurate.

4.4.3 Factor Rotation and Interpretation

The output of factor extraction phase is a factor matrix that contains the weights of each variable on the extracted factors also called loadings. These coefficients are unrotated [58]. They lack meaningful interpretation, although they account for maximum variability. The objective of factor rotation is to preserve the amount the variability while enhancing the interpretability. The improvement is obtained by rotating the axes, thus resulting in a clear separation of variable

clusters on each factor. In other words, each cluster of variables will be associated with one factor which simplifies the interpretation.

There are two methods for factor rotation [56, 57]. Each offers a variety of choices. The orthogonal method, in which the factors are kept at a 90º angle of each other, employs varimax, equimax and orthomax as a rotation method. The oblique method, on the other hand, departs from the right angle rule and offers promax, oblimin and quartimin as options. Orthogonal rotation produces factors that are uncorrelated and easy to interpret. The oblique method takes into consideration the possible existence of correlation between the extracted factors. Forcing orthogonality on correlated factors, results in a loss of information. In exploratory data, it is recommended to implement the oblique method since both methods yield same results on uncorrelated factors. To rotate the factors in our study, we will implement the oblique method with the 'promax' option.

4.4.4 Multiple Regression Analysis

Multiple regression analysis is a statistical tool aimed at uncovering and understanding the relationship between a variable to be explained, called a dependent variable, and explanatory variables, called independent variables [59]. Multiple regression analysis tests whether the independent variables contribute to the variance in the dependent variable or not. It is used in many fields to measure the magnitude of the effect, if present, and forecast its value.

Multiple regression analysis involves many steps, among them, the characterization of the relationship between the dependent variable and the set of predictors by a mathematical representation. The model can adopt a linear or a non-linear form, depending on the nature of the relationship. It is a crucial step to determine this nature before proceeding with the analysis itself. Another step that can substantially bias the results is the choice of the predictors. Failure to select

the appropriate independent variables can lead to the collinearity effect [60]. This happens when two or more independent variables contain much of the same information, making it difficult for regression analysis to determine which one is more important. Another concern, at this step, is the precision of the measurements when irrelevant variables are included, which tend to reduce the precision of the results.

Multiple regression analysis offers a set of tools that helps in the interpretation of the results. The most important are:

- The R-square represents the portion of variance in the dependent variable explained by all the predictors when grouped together.
- The ANOVA (analysis of variance) table, which can be thought of as a significance value for the whole model, confirms the statistical significance of the R-square value.
- The coefficients table indicates both the weight of each variable in the equation and informs about the severity of collinearity.

4.5 Theoretical Analysis

Several experiments were conducted with the statistical approach. The objective is to explore the impact of topology structure on the consensus. In summary, the statistical results described in section 5.2.1 revealed that intra and inter cluster links contribute to some degree in the consensus and synchronization among a set of nodes. The collected data for OCTOPUS communication network emphasized the importance of intra cluster links over inter cluster links in the synchronization process. The subjectivity of these results is a concern. Collecting different data could lead to different results. The question is to prove the validity of our statistical results

and uncover any other condition(s) on the role that intra and inter cluster links play in the consensus.

In this section, we analyze the importance of intra and inter-cluster links in the consensus building for OCTOPUS topology. We proceed by defining a ratio for intra cluster, inter cluster, and overall network synchronization. We investigate the impacts of intra and inter cluster links on these ratios.

The consensus process is a dynamic system by nature. It involves nodes' state update. Some of its features in OCTOPUS underlying lattice are the time and state variables. Modeling the dynamics of OCTOPUS consensus involves defining the so called coupled map lattice which represents the dynamics of a node with respect to the state of other nodes at a given time. The general form of the coupling equation is [61]:

$$s_i(t+1) = f(s_i(t)) + \alpha \sum_{j=1}^{n} a_{ij} w_{ij} [f(s_j(t)) - f(s_i(t))] \ , \tag{28}$$

where $s_i(t)$ is the state of node *i* at time *t*, *n* is the number of nodes, α is the coupling strength, w_{ij} is the weight of the edge between node *i* and *j*, A=[a_{ij}] is the adjacency matrix ($a_{ij} w_{ij} \neq 0$ if $a_{ij} \neq 0$) and *f* is a mapping function.

In this analysis, we focus on a deriving a general form of the coupling equation for intra and inter cluster convergence and define a new condition for synchronization. Similar to section 3.2, we define the communication network as a graph *G=(V,E)* where *V* is the set of vertices or nodes and *E* is the set of edges or links. The set of nodes *V* is divided into *m* separate clusters. $V = \bigcup_{k=1}^{m} C_k$, where C_k is a cluster and $C_k \bigcap C_l = \phi \ for \ k \neq l$. In a cluster, we define *N(i)* as the adjacent neighbors of node *i*. Equation 28 then becomes

$$s_i^k(t+1) = f_k(s_i^k(t)) + \alpha \sum_{k'=1}^{m} \sum_{j=1}^{|C_{k'}|} a_{ij} w_{ij} [f_{k'}(s_j^{k'}(t)) - f_k(s_i^k(t))] \quad (29)$$

Within a cluster *k* the coupling equation is

$$s_i^k(t+1) = f_k(s_i^k(t)) + \alpha \sum_{j=1}^{|C_k|} a_{ij} w_{ij} [f_k(s_j^k(t)) - f_k(s_i^k(t))] \quad (30)$$

We also define $\beta_{ij} = a_{ij} w_{ij}$ which restricts the dynamic of updates to the network organization. For a given cluster, β_{ij} informs about the links between the cluster head, dominant node, and its neighbor(s), dominated node(s). In a cluster, β_{ij} is only dependent of w_{ij} ($a_{ij} = 1$). It can take many forms to represent the cluster lattice. One solution is to make β_{ij} independent of *j* and define it as a ratio of a node degree ($\gamma_i^{intra} = | N_{intra}(i) |$), intra links only, over the highest node degree in any cluster ($\Delta_k^{intra}(G) = \max\{\gamma_i^{intra} : i \in V(G)\}$). Equation 30 becomes

$$s_i^k(t+1) = f_k(s_i^k(t)) + \alpha \sum_{j=1}^{|C_k|} \frac{\gamma_i^{intra}}{\Delta_k^{intra}} [f_k(s_j^k(t)) - f_k(s_i^k(t))] \quad (31)$$

Similarly, we define a coupling equation for nodes that play a role in inter cluster communication. In OCTOPUS, these nodes are the gateways and the cluster heads. The ratio comprises the node degree, in terms of the number of inter cluster links, over the total number of inter cluster links. Equation 31 becomes

$$s_i^k(t+1) = f_k(s_i^k(t)) + \alpha \sum_{k'=1, k' \neq k}^{m} \sum_{j=1}^{|\Gamma|} a_{ij} \frac{\gamma_i^{inter}}{\Delta^{inter}} [f_{k'}(s_j^{k'}(t)) - f_k(s_i^k(t))], \quad (32)$$

where Γ is a set comprising one cluster head and one gateway within a cluster.

To define the Laplacian of the system, we set $l_{ij}^{intra} = \frac{\gamma_i^{intra}}{\Delta_k^{intra}}$ and $l_{ij}^{inter} = a_{ij} \frac{\gamma_i^{inter}}{\Delta^{inter}}$ for intra, respectively inter cluster connection, and $l_{ij} = 0$ otherwise. The Laplacian representing the cluster

connections (intra) will have one row and one column of identical values. One characteristic of the Laplacian matrix is that the sum of each row is equal to zero. For this to hold, we set $l_{ii}^{intra} = -\sum_{j=1}^{|C_k|} \frac{\gamma_i^{intra}}{\Delta_k^{intra}}$ and $l_{ii}^{inter} = -\sum_{j=1}^{|\Gamma|} a_{ij} \frac{\gamma_i^{inter}}{\Delta^{inter}}$. Equations 31 and 32 become

$$s_i^k(t+1) = f_k(s_i^k(t)) + \alpha \sum_{j=1}^{|C_k|} l_{ij}^{intra}[f_k(s_j^k(t)) - f_k(s_i^k(t))] \tag{33}$$

Respectively,

$$s_i^k(t+1) = f_k(s_i^k(t)) + \alpha \sum_{k'=1,k' \neq k}^{m} \sum_{j=1}^{|\Gamma|} l_{ij}^{inter}[f_{k'}(s_j^{k'}(t)) - f_k(s_i^k(t))] \tag{34}$$

The purpose is to derive a convergence ratio and boundary condition(s) for equations 33 and 34. They will help in investigating the impact of intra and inter links on the overall convergence and synchronization process.

When the system approaches the synchronous state $s(t)$, all the nodes exhibit similar state behavior $s_i^k(t+1) = f(s_i^k(t)) = f(s(t))$, which is independent of the state of the neighboring nodes. By denoting $S(t) = [s_1(t),...,s_n(t)]^T$ and $F(S) = [f(s_1),...,f(s_n)]^T$, equations 33 and 34 take the general form:

$$S(t) = (I_n - \alpha L) F(S(t)), \tag{35}$$

where *n* is the number of nodes. Since the sum of each row in the Laplacian matrix is equal to zero, diagonalizing L leads to a set of nonnegative eigenvalues $\lambda_i, i = 1,...n$. This set can be sorted as $\lambda_1 = 0 \leq \lambda_2 \leq ... \leq \lambda_n$. Normalizing L using the corresponding eigenvectors $e_i, i = 1,...n$, results in

$$S(t) = (1_n - \alpha\lambda)^T I_n F(S(t)) \tag{36}$$

At this stage, a small perturbation δs_i of the state is dictated by the equation $\delta s_i(t+1) = f'(s(t))\delta s_i(t)$ [7, 62]. Along the *i*th eigenvector of equation 36, the perturbation is given by:

$$\delta s_i(t+1) = (1-\alpha\lambda_i) f'(s(t))\delta s_i(t) \tag{37}$$

One solution for equation 37 is [7]

$$\delta s_i(t+1) = \delta s_i(t_0)\prod_{l=0}^{t}(1-\alpha\lambda_i) f'(s(l)) \tag{38}$$

A system of more than two nodes reaches a local asymptotic stability if [63, 64, 65]

$$\lim_{t\to\infty}\frac{1}{t}\sum_{l=0}^{t-1}\log\frac{|\delta s_i(l+1)|}{|\delta s_i(l)|} < 0 \quad \Leftrightarrow \quad \lim_{t\to\infty}\frac{1}{t}\log\left|\prod_{l=0}^{t-1}(1-\alpha\lambda_i) f'(s(l))\right| < 0 \tag{39}$$

$$\Leftrightarrow \quad \lim_{t\to\infty}\frac{1}{t}\sum_{l=0}^{t-1}(\log|1-\alpha\lambda_i| + \log|f'(s(l))|) < 0$$

$$\Leftrightarrow \quad \log|1-\alpha\lambda_i| + \lim_{t\to\infty}\frac{1}{t}\sum_{l=0}^{t-1}\log|f'(s(l))| < 0$$

In a dynamical system with an evolution equation, the second term of the addition is identified as the Lyapunov exponent μ [64]. The more negative the exponent, the greater the stability of the system.

$$\mu \equiv \lim_{t\to\infty}\frac{1}{t}\sum_{l=0}^{t-1}\log|f'(s(l))| \tag{40}$$

Equation 39 becomes:

$$|1-\alpha\lambda_i| < e^{-u}, \tag{41}$$

which leads to the following solutions:

$$\lambda_2 > \frac{1-e^{-u}}{\alpha}\ ,\ \lambda_n < \frac{1+e^{-u}}{\alpha}\ \text{and}\ \frac{\lambda_2}{\lambda_n} > \frac{1-e^{-u}}{1+e^{-u}}\ .$$

For any communication network, the smaller the ratio $\frac{\lambda_2}{\lambda_n}$, the better is the convergence and synchronization.

CHAPTER 5

TASK ALLOCATION

5.1 Introduction

The key challenges in accomplishing a task in a multi-robot environment include task allocation and coordination. Fielding multiple robots into an unknown area creates the problem of uncertainty and unpredictability. In a dynamically changing environment, the problem becomes more complex to tackle. While this area of research is still being explored, there are quite a few solutions that have been proposed. A simple and obvious solution is to implement a centralized system that divides the tasks among robots and monitors their progress. Some of the shortcomings of this approach are lack of scalability and reliability. A small increase in the number of robots adds more communication burden and slows down the central entity. A failure of the main entity, responsible for the entire system coordination and resources management, causes a complete shutdown of the system.

The fundamental problem in task allocation is as follows: Given a set of robots and targets (tasks) spread in a geographic area, how can a robot decide on its course of action and contribute to the accomplishment of the overall task? To address the task allocation problem, we propose to compare three distributed techniques that can be implemented on robots. These task allocation strategies are mutually exclusive and do not require any coordination among robots. The objective of this chapter is to explore the robots' behavior when different techniques are employed. Concepts such as target detection, fault tolerance and collision avoidance, et cetera are beyond the scope of this chapter and hence are not considered here.

5.2 Task Allocation Techniques

5.2.1 Kohonen Self-Organizing Map

Self-organizing map (SOM) is a neural network (NN) model that is capable of projecting high-dimensional data onto lower dimensions [66, 67, 68]. The projection is done adaptively and preserves the characteristic features of the data. The algorithm has been successfully used for a number of applications such as density estimation, vector quantization, data visualization, computer-generated music, and pattern recognition among others. The design and development of SOM is inspired from the brain physiology. The inputs are mapped onto different topological areas; thus making the computational map, a crucial information-processing component. Computational maps exhibit the following properties:

- Neurons with similar receptive field profiles are grouped.
- Features of the input data are preserved by the output neurons.

The algorithm responsible for the formation of SOM proceeds by initializing the synaptic weights. A robot's initial coordinates represent its synaptic weights. Once the network has been properly initialized, there are three processes involved in the formation of SOM.

The first process is a competitive process in which the neurons in the map compute their respective values of a discriminant function. The neuron with highest value is declared the winner. Given an input pattern denoted by $Y = [y_1, y_2, ..., y_n]^T$ and a weight vector for a neuron j, $W_j = [w_{j1}, w_{j2}, ..., w_{j3}]$, the winning node is the one that yields the highest inner product $W_j^T Y$. Maximizing the inner product is mathematically equivalent to minimizing the Euclidian distance between Y and W_j.The winning node j is then determined by: $\arg\min_j \| y - w_j \|$.

The second process is a cooperative process which locates the neighbors of the winning node. A firing neuron tends to excite the neurons that are adjacent to it. In order to mimic this excitation in the Kohonen model, a neighborhood function of the lateral distance is defined. A typical choice is a Gaussian function of the form:

$$h_{ij}(x) = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right), \tag{42}$$

where d_{ij} is the distance between node *i* and *j* and σ is the neighborhood radius. The concept of neighborhood is sometimes used to locate a backup and replacement for a failing node.

The third process is a synaptic adaptive process that allows the winning node to adjust its weight in accordance with the Kohonen learning algorithm. The learning is a linear combination of the old weight vector and the current input vector.

$$w_j(new) = w_j(old) - \alpha h(y - w_j(old)) \ , \tag{43}$$

where *y* is the input vector, w_j is the weight vector of unit *j* and α is the learning rate [29]. This weight update algorithm will be used throughout the other methods to move the robot to the assigned target(s).

SOM has a time complexity $\theta(n^2)$, since the Euclidean distance to all neurons on the map is computed for each input pattern (*n* being the number of robots).

5.2.2 Hungarian Method

The second technique explored is the Hungarian technique, which is one of the most popular combinatorial algorithm. Its objective is to solve the assignment problem which is known in graph theory as optimal matching for a bipartite graph. This algorithm is based on ideas of König and Egerváry [70]. It was first introduced by Kuhn and was refined by Munkres

[71, 72]. In the theorem**,** König states that in a bipartite graph, the maximum cardinality of a matching is also the minimum number of nodes covering all the edges.

Generally, given a bipartite graph with a weight function *w*, the cost of a perfect matching K of a graph is defined as $\gamma(\mathrm{K}) = \sum_{e \in K} \gamma(\mathrm{e})$, where the cost *γ(e)* of an edge *e* is defined by

γ(e)= C – w(e), C being the maximal weight of all edges. Thus, determining the optimal matching in a graph with respect to the weight function is equivalent to solving the assignment problem for the matrix M where $m_{ij} = C - w_{ij}$ with w_{ij} being the weight of edge between nodes i and *j*. This is equivalent to finding the perfect matching of minimal cost.

The algorithm requires a square matrix and starts by converting the problem of maximum assignment to a problem of minimum assignment by replacing each w_{ij} *by* $C - w_{ij}$. It is made up of three parts. The first two steps of the algorithm generate a matrix with at least one zero in each row and column. The last step generates an optimal zero assignment through an iterative process. Initially, the brute force algorithm for solving the assignment problem has a time complexity *θ(n!)* since for a given *n x n* matrix, there are n possible solutions for the first target, *n-1* possible solutions for the second target and so on. The Hungarian algorithm, as proposed by Kuhn, requires three nested loops, thus it has a running time $\theta(n^3)$ which is polynomial compared to the exponential or factorial. The fastest flavor of the Hungarian algorithm was proposed by Gabow and has a complexity $\theta(n^2 \log n)$ [69].

5.2.3 Linear Programming Method

Linear programming (LP) is among the most widely used tool in decision making [39, 72]. Such problems arise in all walks of life. Whether in economics, strategic planning, algorithms analysis, production scheduling and planning or train scheduling, the tool's aim is to

transform a real-life practical problem into a mathematical model. It consists of constructing an objective function that needs to be maximized or minimized, specifying a set of constraints and defining a finite number of variables. In the case where the variables are restricted to integers, the linear model is called an integer linear programming (ILP) model. In our case, the decisions are limited to 1 (assigned to target) or 0 (not assigned). Translating the assignment problem for *n* robots and *m* targets into a formulation requires the following definitions:

The objective is to find a minimum cost assignment.

$$\begin{aligned}
&\text{minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij}. \quad i=1,\ldots,n.\ j=1,\ldots,m. \\
&\textit{subject to} \\
&\quad \sum_{j=1}^{m} x_{ij} \le b_i \qquad b_i = \begin{cases} 1 & \textit{if } n \ge m. \\ ceil\left(m \div n\right) & \textit{otherwise.} \end{cases} \\
&\quad \sum_{i=1}^{n} x_{ij} = 1 \\
&\quad x_{ij} = \begin{cases} 1 & \textit{if robot } i \textit{ is assigned to } \text{target } j, \\ 0 & \textit{otherwise} \end{cases}
\end{aligned} \tag{44}$$

where c_{ij} is the cost for robot *i* to service target j, b_i is the maximum number of targets a robot can service and x_{ij} is the variable to be evaluated. The first constraint assigns more targets to a robot in case the number of targets exceeds the number of robots. The second constraint refers to indivisible tasks.

Many exhaustive and implicit enumeration methods have been proposed to solve this type of optimization problem. The most popular are branch and cut, branch and bound, cutting plane, Dantzig-Wolf decomposition simplex, et cetera [67]. For our ILP problem, we propose a branch and bound based solution due to the popularity of the technique. Basically the branch and bound method uses a tree structure to solve the problem. It partitions its feasible region into subsequent smaller subsets (branching). It then calculates bounds on the objective function for

each subset (bounding). The bounds are, then, used to discard some subsets (fathoming) and to update the current solution by a better solution (relaxation). The iterative process stops when no feasible solution or better solution could be produced.

In this study, two search techniques were considered: best-node and depth-first. Best-node traverses the tree level by level visiting all the nodes at the top level before searching the bottom level and has a time complexity $\theta(2^d)$ d is the depth level of the tree. Depth-first search traverses the tree branch by branch going all the way to the bottom leaf. It has similar time complexity but less space requirement to execute. During simulation, the focus was on the best-node search since it guaranties an optimal solution while depth-first stops at local minimum.

5.3 Theoretical Analysis

So far, we presented the three techniques. An explorative simulation was conducted to investigate the behavior of the three methods (see details in section 6.3.1). In summary, the preliminary simulation revealed that the three techniques generate different assignment solutions. However, for some configurations the assignment solutions are identical. The experiments revealed, also, that the ILP and Hungarian methods are almost similar in terms of cost of assignment and workload distribution. This result is predictable, considering that both methods are two optimization techniques. As a result, the goal of this section is to try to reproduce some of the results obtained in the explorative simulation. The idea is to compare SOM to one combinatorial optimization technique ILP or Hungarian. The focus is on the workload distribution through a fairness ratio and the cost of assignment of the selected robots. SOM is an NN. In order to compare SOM to an LP optimization technique, we must design a suitable common representation for both techniques. A model of differential equations is developed to

represent and capture the dynamics of the two techniques. A validation phase of the developed model is also presented before reproducing some of the results.

5.3.1 ILP optimization problem

5.3.1.1 System Dynamics

The objective of the ILP or Hungarian implemented method is to find a task assignment plan for the robots by solving the following LP problem.

$$\begin{aligned} &\text{minimize} \quad f(x) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \\ &\textit{subject to} \\ &\qquad r_{i1}(x) = \sum_{j=1}^{m} x_{ij} - b_i \le 0 \\ &\qquad r_{j2}(x) = \sum_{i=1}^{n} x_{ij} - 1 = 0 \\ &\qquad r_{i3}(x) \Leftrightarrow x_{ij} \in \{0,1\}, \end{aligned} \tag{45}$$

where *n* is the number of robots and *m* is the number of targets. Here, it is important to mention that the Hungarian method can be expressed as an LP problem by negating $r_{j2}(x)$ constraint [74]. The key step in designing an adequate NN for any linear problem is the construct of an appropriate energy or cost function *E(x)* [75]. One way of selecting a desired energy function is by defining a penalty function. The goal is to transform a constraint LP formulation to an unconstraint one by adding a penalty term to the objective function. The purpose of the penalty function is to inform about the amount by which the constraint is violated.

Equation 43 could be transformed into the following general optimization problem:

$$\text{minimize} \quad E(x) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{p=1}^{l} \alpha_p P[r_p(x)] , \tag{46}$$

where $\alpha_p > 0$ is the penalty parameter. $P[r_p(x)] = \begin{cases} 0 & if\ r_p(x)\ is\ verified \\ 1 & otherwise \end{cases}$ is the penalty term. Equation 46 is equivalent to:

$$\text{minimize} \quad E(x) = f(x) + \alpha_1 \sum_{i=1}^{n} P[r_{i1}(x)] + \alpha_2 \sum_{j=1}^{m} P[r_{j2}(x)] + \alpha_3 \sum_{i=1}^{n} P[r_{i3}(x)] \tag{47}$$

Furthermore, the penalty term could take different forms. Each expression requires an accompanying proof of the accuracy of the approximation. In this study, we express the penalty term as: $P[r_{ik}(x)] = [\min\{0, r_{ik}(x)\}]$.

This definition offers the advantage of being "exact" in the sense that the minimum of *E(x)* is equivalent exactly to equation 43 solution [75, 76]. If the constraint condition is verified, then the $r_{ik}(x)$ signal is inhibited, otherwise the signal is amplified by α and passed to the next layer of the network. Equation 47 then becomes:

$$\begin{aligned} &\text{minimize} \quad E(x) = f(x) + \alpha_1 \sum_{i=1}^{n} \min\{0, r_{i1}(x)\} + \alpha_2 \sum_{j=1}^{m} \min\{0, r_{j2}(x)\} + \alpha_3 \sum_{i=1}^{n} \min\{0, r_{i3}(x)\} \\ &\Leftrightarrow f(x) + \alpha_1 \sum_{i=1}^{n} \min\{0, \sum_{j=1}^{m} x_{ij} - b_i\} + \alpha_2 \sum_{j=1}^{m} \min\{0, \sum_{i=1}^{n} x_{ij} - 1\} + \alpha_3 \sum_{i=1}^{n} \sum_{j=1}^{m} \min\{0, x_{ij}(1 - x_{ij})\} \\ &\Leftrightarrow f(x) + \alpha_1 \sum_{i=1}^{n} \min\{0, \sum_{j=1}^{m} x_{ij} - b_i\} + \alpha_2 \sum_{j=1}^{m} \min\{0, \sum_{i,k=1, i \neq k}^{n} x_{ij} x_{kj}\} + \alpha_3 \sum_{i=1}^{n} \sum_{j=1}^{m} \min\{0, x_{ij}(1 - x_{ij})\} \end{aligned} \tag{48}$$

The first term of the energy function in equation 48 denotes the original cost. The second term represents the constraint that a robot can service up to *b* targets. The third term specifies that each target requires only one robot to service it. And the last term forces the output x_i to be 0/1.

The question now is how to solve this unconstraint problem with an energy function containing a penalty term. Equation 46 form suggests that the search for a solution should be performed through an iterative process similar to [75]:

$$X^{(k+1)} = X^k + \eta^k d_k, \quad X(0) = X^{(0)} \quad (k = 0, 1, 2...) \,, \tag{49}$$

where η^k denotes the length of the step to be taken in the direction of the vector d_k.

Different techniques are proposed to compute η^k and d_k. In this study we select he steepest-descent (gradient) method, and define the direction $d_k = -\nabla E(X^{(k)})$. Equation 48 could be mapped to this system of differential equations [75]:

$$\begin{aligned} &\frac{dx}{dt} = -\mu \nabla_x E(x), \quad x(0) = x^{(0)} \\ &\mu = diag(\mu_1, \mu_2, \mu_3, ..., \mu_n) \quad \mu_i > 0 \end{aligned} \tag{50}$$

In its scalar form for any target *j*:

$$\left(\begin{aligned} &\frac{dx_{1j}}{dt} = -\mu_1 \left(\frac{\partial f(x)}{\partial x_{1j}} + \sum_{p=1}^{l} \alpha_p \frac{\partial P}{\partial r_p} \frac{\partial r_p(x)}{\partial x_{1j}} \right), \quad x_{1j}(0) = x_{1j}^{(0)} \\ &\frac{dx_{2j}}{dt} = -\mu_2 \left(\frac{\partial f(x)}{\partial x_{2j}} + \sum_{p=1}^{l} \alpha_p \frac{\partial P}{\partial r_p} \frac{\partial r_p(x)}{\partial x_{2j}} \right), \quad x_{2j}(0) = x_{2j}^{(0)} \\ &. \\ &. \\ &. \\ &\frac{dx_{nj}}{dt} = -\mu_n \left(\frac{\partial f(x)}{\partial x_{nj}} + \sum_{p=1}^{l} \alpha_p \frac{\partial P}{\partial r_p} \frac{\partial r_p(x)}{\partial x_{nj}} \right), \quad x_{nj}(0) = x_{nj}^{(0)} \end{aligned} \right) \tag{51}$$

5.3.1.2 Convergence

It is important to ensure that the cost function *E(x)* will converge to a local minimum after a specified period of time. This suggests that the cost function will enter the neighborhood of the minimum and remain there. In order to verify this aspect, let's take the time-derivative of the energy function after normalizing equation 50 and assuming that μ is a positive time-variable parameter. The goal is to show that the cost function decreases as time progresses.

Normalizing and setting μ = μ (t) yields to the general form:

$$\frac{dx}{dt} = -\mu(t)\frac{\nabla_x E(x)}{\left\|\nabla_x E(x)\right\|^2}, \quad where \quad \left\|\nabla_x E(x)\right\|^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{\partial E}{\partial x_{ij}}\right)^2 \tag{52}$$

$$\frac{dE}{dt} = [\nabla_x E(x)]^T \frac{dx}{dt} = -\mu(t)\frac{[\nabla_x E(x)]^T \nabla_x E(x)}{\left\|\nabla_x E(x)\right\|^2} = -\mu(t) < 0 \tag{53}$$

If we consider that $\frac{dE}{dt} = 0$ iff $\nabla_x E(x) = 0$ it follows that the trajectory of the cost function is a strictly monotonically decreasing time-function. It only depends on μ(t).

5.3.1.3 Fairness

Measuring the fairness in a task assignment problem is another metric that need to be investigated using the LP differential equations form. It will inform about the workload balancing and allow the comparison between LP and SOM. It is easy to verify it under ILP formulation since the fourth term of the energy function informs about the selection or not of a robot. We define the ILP fairness index for a given configuration as:

$$F_{ILP} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}x_{ij}}{\left(\sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}\right)^2} \quad x_{ij} \in \{0,1\}, \tag{54}$$

where $C_{ij} \neq c_{ij}$ is the evaluated cost of executing a task using equation 51.

5.3.2 Self-Organizing Map

5.3.2.1 System Dynamics

In order to compare the two methods, we need to define a set of differential equations which models the behavior of nodes when SOM is employed. Presented by an input ξ, SOM's learning algorithm (equation 43) could be formulated as the following:

$$w_i(t+1) = w_i(t) + \alpha(t).h(\xi, w_i(t)), \tag{55}$$

where h is the neighborhood function. In its vector form, equation 53 becomes:

$$\vec{w}_{t+1} = \vec{w}_t + \alpha_t.\vec{H}(\xi, \vec{w}_t). \tag{56}$$

Equation 55 is similar to a recursive algorithm which can be associated with the following ordinary differential equations (ODE) [77]:

$$\vec{w}^D = \vec{h}(\vec{w}), \tag{57}$$

where $\vec{h}(\vec{w})$ is the mean vector field and can be solved using Euler approximation method for ODEs [77].

5.3.2.2 Convergence

As the step size $h \to 0$, Euler's numerical method solution approaches the exact solution. Different mathematical approaches were proposed to show that Euler' numerical scheme is stable and consistent thus convergent [77, 78].

5.3.2.3 Fairness

Measuring fairness under Euler formulation requires detecting the active robots. The only way to select a robot is through its lower cost to service a target T. An example will be provided in the simulation. Similarly, we define a fairness index for SOM formulation as:

$$F_{SOM} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}x_{ij}}{\left(\sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}\right)^2}$$
$$x_{ij} == \begin{cases} 1 & if \quad i = \arg\min_k((C_{kj}(T_j)), \\ 0 & otherwise \end{cases} \tag{58}$$

where C_{ij} is the cost evaluated using Euler method on equation 57.

CHAPTER 6

RESULTS

6.1 Topology Creation

The proposed on-demand topology update strategy is simulated in MATLAB. In this section, the network topology formation is explained step by step. To test our approach, a series of experiments have been conducted with different sets of nodes that were randomly placed in a geographic area. Figure 6 shows the results of cluster formation, cluster-head election and gateways selection of eight nodes with equal transmission range. In this figure, four clusters were identified. Cluster 1 comprises nodes CH7, N1, N4 and G6. Cluster 2 is formed by CH5 and G3. Finally, CH8 and CH2 form clusters on their own. A node labeled by CH is the elected cluster-head in each cluster, a node labeled by G is the gateway, and a node labeled N is a normal node. In the case of single node clusters, the cluster-head plays the role of a gateway as well. In Figure 6, CH7 and CH5 are cluster-heads that use gateways G6 and G3 respectively.

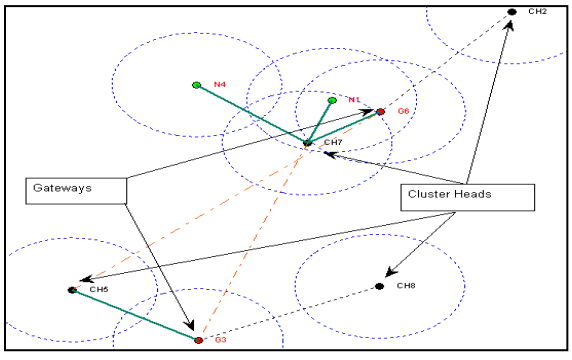

Figure 6: Cluster-head and gateway selection with four clusters.

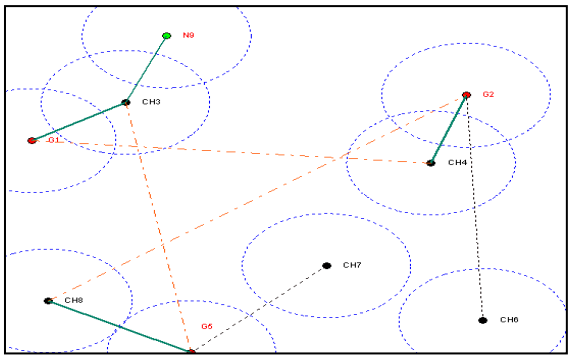

Figure 7: Connecting multiple cluster-heads and gateway.

Isolated clusters need to establish links among themselves. As a result of applying equation 17, a link is established between a single-node cluster and a gateway. CH2 is assigned

to G6 and CH8 is connected to G3. Figure 6 illustrates the establishment of the links between the single-node clusters and the gateways.

The final step is to establish a link between the multi-node clusters via their respective gateway. Using equation 16, CH5 is linked to G6 and CH7 is connected to G3. Figure 6 illustrates the establishment of the link between multi-node clusters. However, contrary to the connections shown in Figure 6, not all gateways will be connected to all cluster-heads and vice versa. The implementation is such that an optimal connection is established among the set of gateways and cluster-heads. Figure 7 shows a fault tolerant connection between three cluster-heads and three gateways. No link between CH3 and G2 or CH4 and G5 is established.

There are factors that could affect the overall structure of the communication network: the number of nodes and the transmission range. Figure 8 shows that increasing the number of nodes lead to an increase, followed by a decrease of the number of clusters. In general, by saturating the area with a higher number of nodes, the number of clusters tends to get smaller. This is illustrated in Figure 10, twenty nodes with equal transmission range form one cluster having CH1 as a cluster-head and G12 as a gateway. This configuration has a negative impact on the power and workload of the cluster-head. Assuming that the probability of a link or node failure is minimal, the call for a topology reconfiguration will be mainly a result of low battery power of the cluster-head.

On the other hand, Figure 9 shows that reducing the transmission range increases the number of clusters. As illustrated in Figure 11, the same twenty nodes with lower transmission range resulted in the appearance of fourteen clusters and three gateways (G18, G5 and G9). Compared to Figure 10, this configuration requires more processing and computing time to establish links between all the clusters.

Another experiment involves the distribution of intra-cluster and inter-cluster links established between the nodes. The purpose of this experiment is to measure the quality of the clustering technique and the goal is to obtain dense intra-cluster links and sparse inter-cluster links. Figure 12 shows that the ratio of intra-cluster links to inter-cluster links increases with an increase in the transmission range. Figure 13 demonstrates a similar result with respect to an increase in the number of nodes in the network.

Since the nodes are positioned randomly, there are situations where all nodes form clusters on their own. In those cases, a single link is established between the cluster-heads. Figure 14 shows links establishment between single-node clusters. The major concern in these topologies is the cost of a link failure. Any loss of connection will automatically require a topology reconfiguration as opposed to a link failure between a gateway and a multi-node cluster. Figure 7 shows that no topology update is required if a multi-node cluster loses one of its inter-cluster links.

Dynamic transmission range is another factor that affects the structure and the role of the node in a network. Figure 15 shows similar nodes positions presented in Figure 7 with different transmission ranges. The result is a new communication network composed of four clusters in which some nodes assume different role. Node 1 and 7 were respectively a gateway and a cluster head in figure 7. In Figure 15 they become respectively a normal and a gateway node. Testing the capabilities of the approach with random positions and random transmission ranges was very tedious and challenging. Thus, the necessity of keeping the transmission range parameter as constant for all nodes.

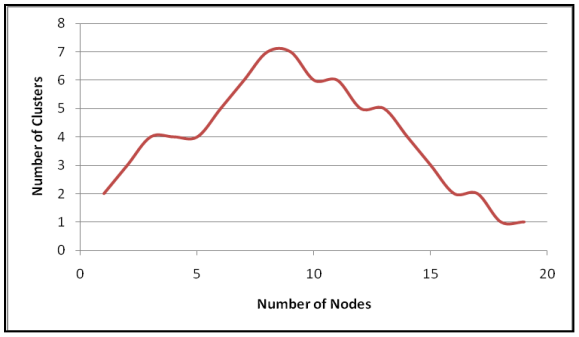

Figure 8: Relationship between the number of clusters and the number of nodes.

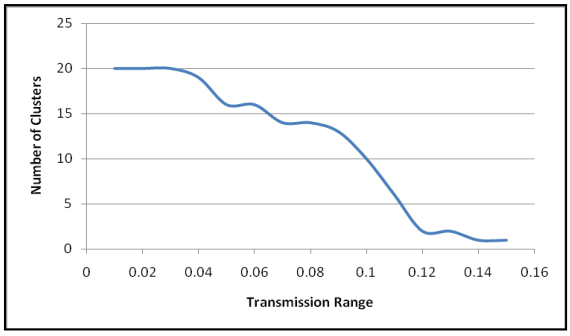

Figure 9: Effect of increasing the transmission range on the number of clusters.

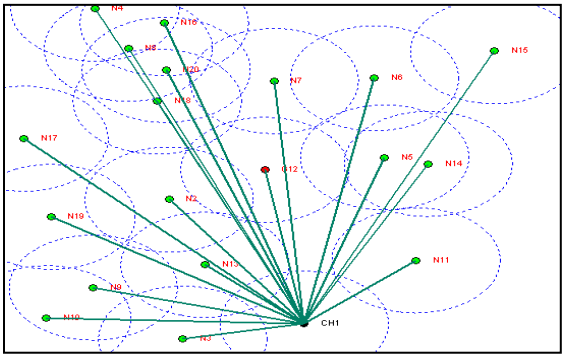

Figure 10**:** Formation of one cluster due to a higher number of nodes.

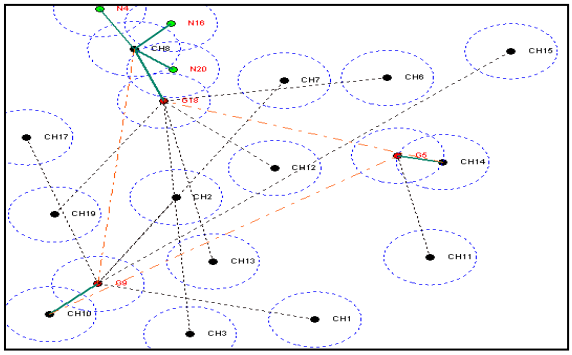

Figure 11**:** The number of cluster increases when the transmission range is reduced.

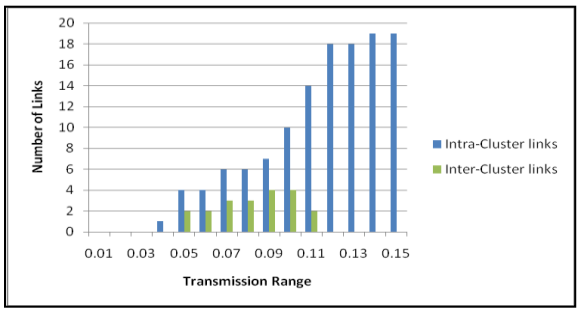

Figure 12**:** Transmission range versus density of intra-cluster and inter-cluster links.

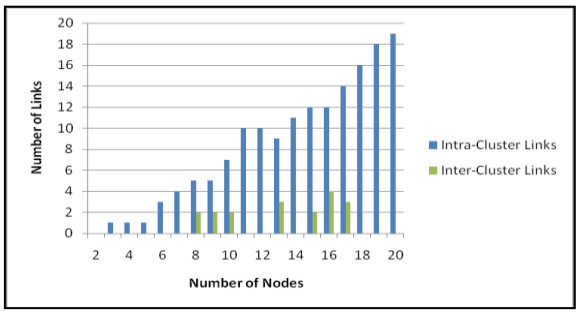

Figure 13**:** Number nodes in the network versus density of intra-cluster and inter-cluster links.

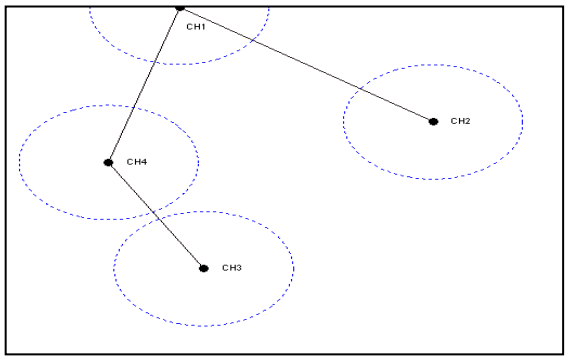

Figure 14: Linking single-node clusters

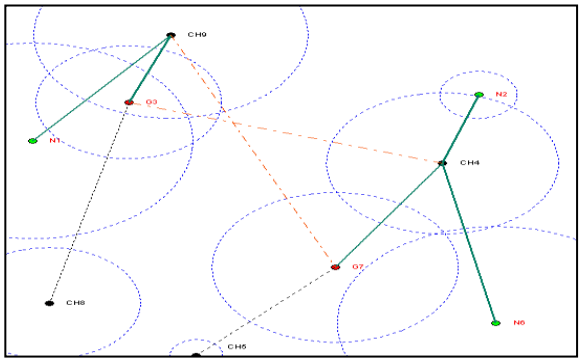

Figure 15: Connecting nodes with different transmission range.

To compare the performance of our technique (OCTOPUS) in terms of clustering capabilities, we conducted two scenario experiments similar to the one proposed by Chen et al [79].Three different clustering techniques were selected: Hierarchical, subtractive and quality threshold clustering.

Hierarchical clustering relies on an agglomerative algorithm to find clusters [80]. It considers each node as a single cluster and combines them to find the final number of clusters. Subtractive clustering computes the likelihood of a node being a cluster head [80]. The nodes in its vicinity are considered neighbors. Quality threshold clustering requires defining a diameter for a cluster such as the maximal distance between two nodes [80]. It relies on adding a candidate node that minimizes the increase of cluster diameter. None of these techniques requires defining the number of clusters in advance.

The experiments consist of increasing the number of randomly placed nodes in a small area and applying the four clustering techniques. The number of clusters and the average cluster size were the two parameters selected to describe the clustering capabilities. OCTOPUS exhibited a better clustering behavior. In Figure 16 OCTOPUS, consistently, maintains a lower number of clusters compared to the other techniques. Its curve indicates that the number of clusters tend to decreases when the number of nodes increases as mentioned previously. In Figure 17 OCTOPUS tends to merge small clusters and favors big cluster sizes while the others create clusters of smaller sizes.

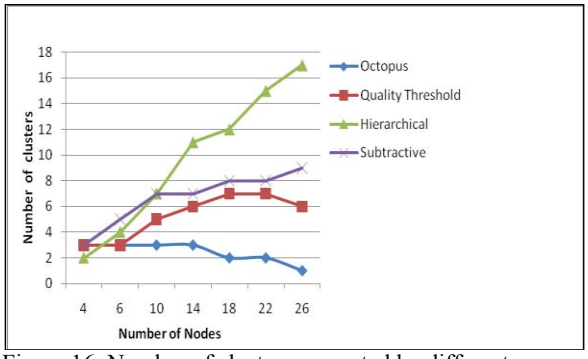

Figure 16: Number of clusters generated by different techniques.

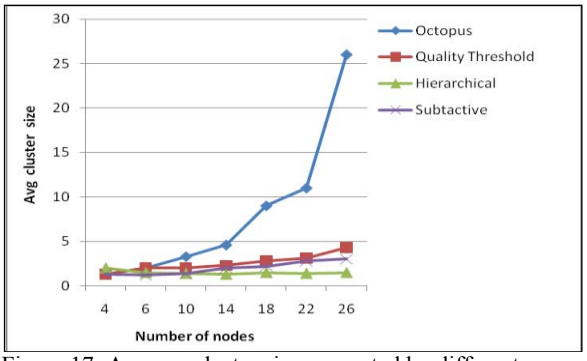

Figure 17: Average cluster size generated by different techniques.

Although OCTOPUS outperforms the quality threshold technique, their curves in Figure 16 suggest that they both exhibit similar behavior. Given appropriate diameter value, both techniques could be similar. Figure 18 illustrates the behavior of quality threshold technique when implemented with different diameter values. As the diameter increases, the quality threshold approaches the OCTOPUS behavior. Thus, OCTOPUS clustering technique could be assimilated to a quality threshold clustering technique with a dynamic diameter adjustment.

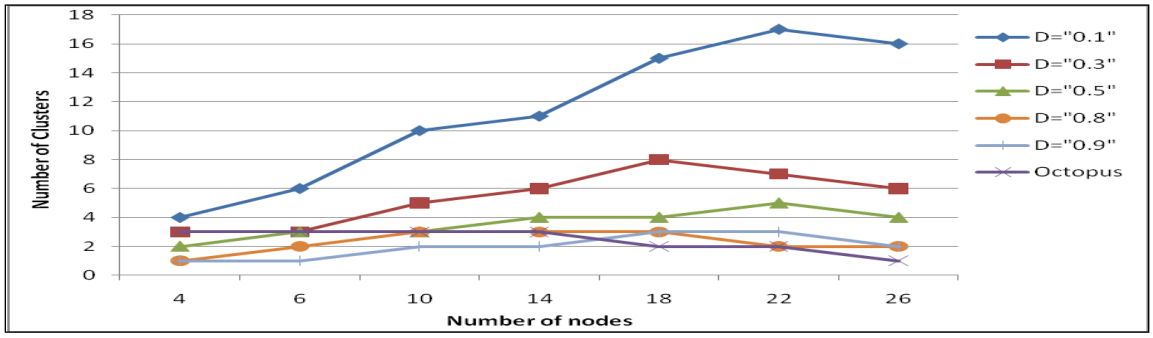

Figure 18: Quality threshold behavior with different diameter values.

A comparative experiment was conducted to evaluate the overall behavior of our deterministic approach against a probabilistic approach. Barolli et al. introduced a probabilistic cluster head election technique based on the available power, node degree and distance to the cluster center [81]. Zhang et al. proposed a gateway selection based on a probability *p* [82].

Combining the two approaches resulted in complete probabilistic selection technique for gateways and cluster heads. Scores of random topologies of 10 nodes were generated and two parameters were evaluated using OCTOPUS and the probabilistic approach. The parameters are the number of cluster heads and the cluster size. The simulation revealed that OCTOPUS generated less cluster heads compared the probabilistic technique. Figure 19 shows the number of cluster heads created by the probabilistic approach most of the time exceeds OCTOPUS's number of cluster heads. This is a result of the creation of more redundant cluster heads in a cluster since many cluster heads candidates satisfy the probability condition. This problem was encountered in OCTOPUS but with less degree. The experiment showed that the introduction of extra variables in addition to the node degree increases the pool of cluster head candidates. The second parameter to be evaluated is directly related to the number of cluster heads. An increase in cluster number leads to a smaller cluster size. The cluster size generated by OCTOPUS is often higher than the cluster size created by the probabilistic approach. This is depicted in Figure 20. In general, OCTOPUS technique tends to optimize by accommodating the maximum number of nodes under the minimum number of clusters.

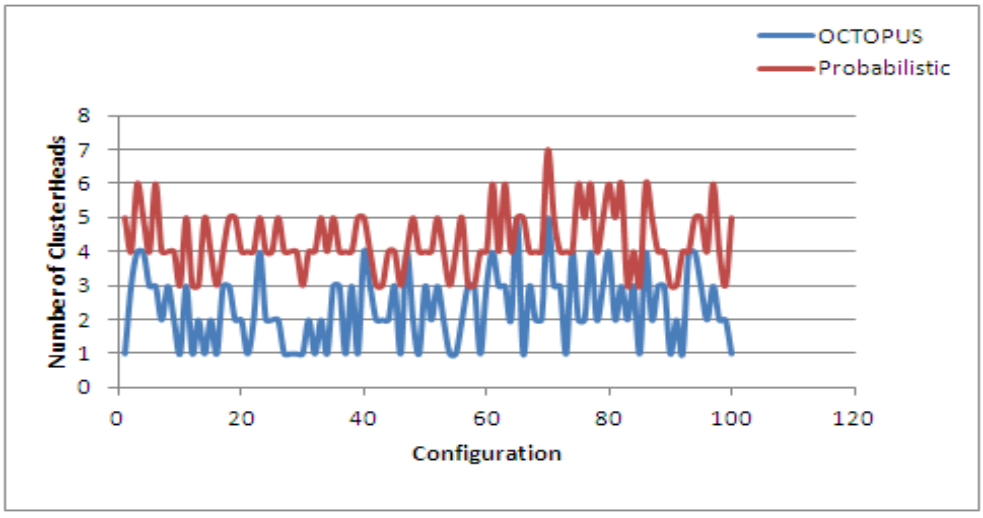

Figure 19: Number of cluster heads generated by the two techniques.

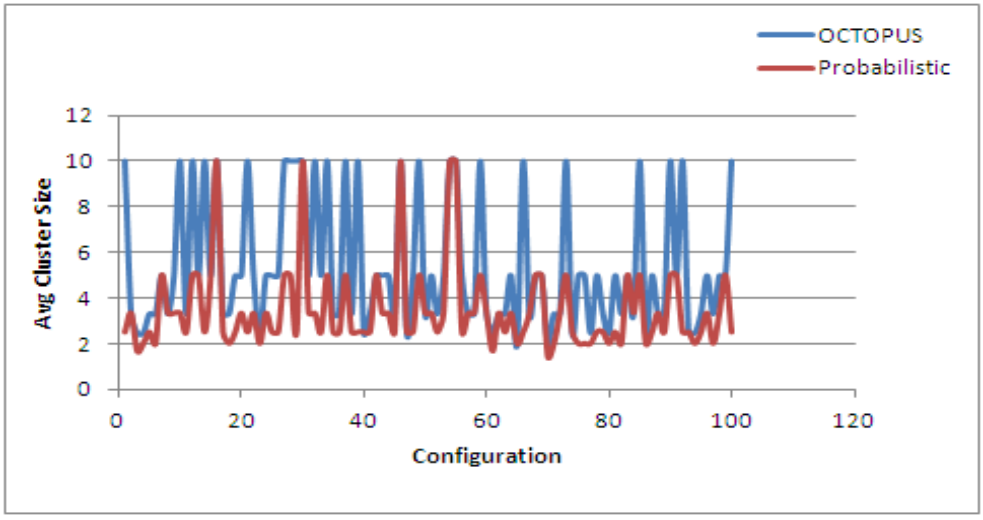

Figure 20: Cluster size generated by the two techniques.

6.2 Topology Impact on Consensus

6.2.1 Statistical Results

Series of experiments were conducted in a MATLAB environment where scores of communication topologies were generated using 8 robots randomly dispersed in a small geographic area. The choice of 8 robots was dictated by the desire of obtaining a balance between the number of single-node clusters and multiple-node clusters. The number of targets to be assigned to the robots was also fixed to 8. Hopping that in the best case the initial scope of each robot will be 1. The coordinates of targets, however, remained unchanged, thus allowing one parameter (topology) to change while targets positions are fixed.

Collecting data was performed in two phases. First, the six structural metrics were evaluated for each topology generated. The same topology was then used, in the second stage, as a communication mean for the robots to update their target scope and evaluate the average state update. The data was then fed to the Statistical Package for the Social Sciences (SPSS) for factor analysis and regression modeling.

TABLE 3(a)

PERCENTAGE CONTRIBUTION OF EACH VARIABLE TO THE TOTAL VARIANCE OF A VARIABLE

Topology metrics	1	2	3	4	5	6	Total
Algebraic Connectivity Ratio(1)	~	4.7	0.3	4.6	10.6	13	39
Average Path (2)	4.7	~	0.2	11.1	4	0.8	14
Average Cluster Coefficient (3)	0.3	0.2	~	0.1	0	1.5	3.7
Average Matching Index (4)	4.6	11.1	0.1	~	36.7	27	62
Modularity (5)	10.6	4	0	36.7	~	4	41
Average Participation (6)	13	0.8	1.5	27	4	~	52
Total	39	14	3.7	62	41	52	

TABLE 3(b)

PERCENTAGE CONTRIBUTION OF EACH VARIABLE TO THE TOTAL VARIANCE OF A VARIABLE. THE SHADED AREAS ARE THE CONTRIBUTION THAT FAILED THE STATISTICAL TEST OF SIGNIFICANCE. THE LAST COLUMN INDICATES THE COMBINED CONTRIBUTIONS OF ALL VARIABLES TO THE VARIANCE OF ONE VARIABLE

Topology metrics	**1**	**2**	**3**	**4**	**5**	**6**	**Total Contribution**
Algebraic Connectivity Ratio (1)	~	L	L	L	L	L	Medium
Average Path (2)		~	L	L	L	L	Low
Average Cluster Coefficient (3)			~	L	L	L	Low
Average Matching Index (4)				~	M	L	Medium
Modularity (5)					~	L	Medium
Average Participation (6)						~	Medium

After normalizing the data, the next step in the methodology described in section 4.4.1 is the evaluation of the correlation matrix for the six metrics. Using the coding listed below, table 3a summarizes the contribution percentage of each variable to the overall variance of a topology metric. The cases that failed the statistical test of significance are represented by the shaded area. Since the matrix is symmetric, only the upper diagonal is presented.

- Low (L) : $0\% \leq v < 30\%$;
- Medium (M) : $30\% \leq v < 70\%$;
- High (H) : $70\% \leq v < 100\%$;

From table 3b, all variables exhibit little influence over each other variance. Thus, suggesting the existence of a weak correlation between the variables. However, the last column suggests that the variability in one variable can be explained by the combined variance in all other metrics. Table 3a, also, reveals that average cluster coefficient failed the test of significance suggesting that its contribution is minimal if not inexistent.

Combining factor extraction and rotation stage, maximum likelihood and principal components analysis, with 'promax' rotation, were conducted to assess the underlying structure for the six variables. Several assumptions were tested, beforehand, to verify the existence of a factor analytic solution. The Kaiser-Meyer-Olkin and Bartelett's tests were performed to measure the sampling adequacy and spehericity [60]. The first test reveals whether the distribution of data is acceptable for conducting a factor analysis. The level attributed to the data was on the border between middling and mediocre. The second test measures whether data is normally distributed and correlated enough to provide a reasonable basis for factor analysis. This condition was also met since the significance value for the test was far below the 0.05 value required. When experiencing with factor extraction, maximum likelihood did not allow the extraction of more than two factors. The two extracted factors were designed to represent two constructs: intra-cluster structure (factor 1) and inter-cluster structure (factor 2). After rotation the first factor accounted for 35% of variance and the second factor for 24%. Table 4 displays the variables and factor loadings for the rotated factors along with their communalities (C) which represent the amount of variance in each variable accounted for by the factor. A high value indicates that the extracted factors are a good representation of the variables. Values less than 0.5 were omitted to improve clarity.

TABLE 4

LOADINGS FOR EACH FACTOR EXTRACTED, THE COMMUNALITIES VALUES AND THE VARIANCE ACCOUNTED FOR. *NOTE*. LOADINGS AND COMMUNALITIES (C) < .5 ARE OMITTED

Methods / Factors / *Variables*	**Maximum Likelihood**			**Principal Components**		
	Factor Loading					
	1	2	**C**	1	2	**C**
Algebraic Connectivity Ratio		.65	.52		.76	**.79**
Average Path				.55		
Average Cluster Coefficient						
Average Matching Index	-.91		.87	-.88		**.83**
Modularity	.68			.80		**.65**
Average Participation		.78	.84		.85	**.86**
Eigenvalues	2.1	1.5		2.1	1.5	
% of variance	**35**	**24**		**35**	**24**	

Few observations can be made from table 4. The communalities evaluated by the principal components indicate that the factors extracted represent well the original variables. Both methods reveal that the average cluster coefficient is not a determinant variable in variance changes. Another observation, in which both methods differ, is related to the average path variable. Principal components report it as a variable to be considered with a low loading whereas maximum likelihood suggests ignoring it.

In a second phase, a multiple regression assessment was conducted in order to determine the best linear combination of the five structural variables for predicting the consensus variable (average state update). Prior to the assessment, there are some assumptions that need to be considered. The linear regression requires the relationship between each of the independents to the dependent variable to be linear. There are different ways to evaluate the linearity. The graphical method uses scatter-plots to visualize the relationship. It is the commonly recommended although difficult to interpret when the sample size is large. Another technique, statistical method, relies on the test of significance of the linearity hypothesis. The relationship is linear if the correlation coefficient between the independent variables and the dependent

TABLE 5

SIGNIFICANCE TEST OF LINEARITY VERIFIED FOR ALL THE INDEPENDENT VARIABLES. P<0.05

Correlations

		Average_ Update	Algebraic_ Connectivity_ ratio	Average_Path	Average_ Matching_ Index	Modularity	Average_ Participation
Pearson Correlation	Average_Update	1.000	-.155	.121	-.351	.472	.497
	Algebraic_Connectivity_ ratio	-.155	1.000	-.217	.218	-.328	.360
	Average_Path	.121	-.217	1.000	-.334	.198	.092
	Average_Matching_Index	-.351	.218	-.334	1.000	-.305	-.519
	Modularity	.472	-.328	.198	-.305	1.000	.200
	Average_Participation	.497	.360	.092	-.519	.200	1.000
Sig. (1-tailed)	Average_Update	.	.000	.000	.000	.000	.000
	Algebraic_Connectivity_ ratio	.000	.	.000	.000	.000	.000
	Average_Path	.000	.000	.	.000	.000	.002
	Average_Matching_Index	.000	.000	.000	.	.000	.000
	Modularity	.000	.000	.000	.000	.	.000
	Average_Participation	.000	.000	.002	.000	.000	.
N	Average_Update	1000	1000	1000	1000	1000	1000
	Algebraic_Connectivity_ ratio	1000	1000	1000	1000	1000	1000
	Average_Path	1000	1000	1000	1000	1000	1000
	Average_Matching_Index	1000	1000	1000	1000	1000	1000
	Modularity	1000	1000	1000	1000	1000	1000
	Average_Participation	1000	1000	1000	1000	1000	1000

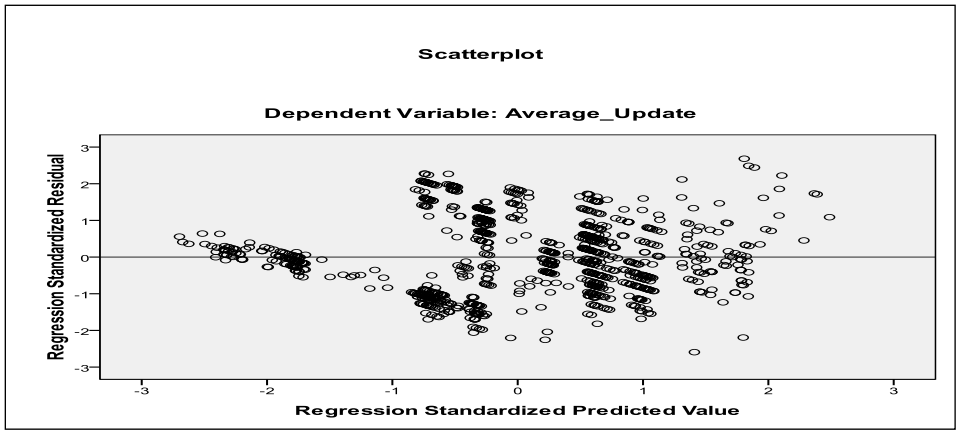

Figure 21: Residual scatter-plot indicating that the errors are normally distributed and no pattern is generated.

TABLE 6

REGRESSION MODEL PREDICTION OF THE CONSENSUS BUILDING VARIABLE

ANOVA[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	76.975	2	38.487	498.550	.000[a]
	Residual	76.967	997	.077		
	Total	153.942	999			

a. Predictors: (Constant), intra-cluster, Inter-cluster

b. Dependent Variable: Average_Update

variable is statistically significant. This condition was met for all the five considered variables. Table 5 shows that the tests are below the 0.05 level of significance. Another assumption that was also checked is the residual (predicted minus observed values) being normally distributed. The scatter-plot in Figure 21 shows that dots are scattered and no patterns are created.

TABLE 7

THE ADJUSTED R SQUARE INDICATED THE PRESENCE OF A FAIRLY GOOD MODEL EXPLAINING 50% OF THE VARIANCE IN THE CONSENSUS BUILDING VARIABLE

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.707[a]	.500	.499	.277847

a. Predictors: (Constant), intra-cluster, Inter-cluster

TABLE 8

THE IMPORTANCE OF INTRACLUSTER PARAMETERS IN THE PREDICTION. THE COLLINEARITY EFFECT IS MINIMAL SINCE TOLERANCE IS > 0.5= (1-0.499) AND THE MODEL COULD BE REPRESENTED BY THE FOLLOWING EQUATION

AVERAGE-STATE-UPDATE = 0.271*INTRA-CLUSTER + 0.135*INTER-CLUSTER + 1.817

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	Collinearity Statistics	
		B	Std. Error	Beta			Tolerance	VIF
1	(Constant)	1.817	.009		208.821	.000		
	Intra-cluster	.271	.009	.648	28.917	.000	.998	1.002
	inter-cluster	.135	.010	.311	13.880	.000	.998	1.002

a. Dependent Variable: Average_Update

The ANOVA analysis of the regression model, in table 6, showed that the combination of variables significantly predict the average update (sig<0.05). The model summary in table 7 exhibited a multiple correlation coefficient R equal to 0.71 and the adjusted R-square is 0.5 meaning that 50% of the variance in average state update variable can be predicted by algebraic connectivity ratio, average matching index , modularity and average participation combined. According to Cohen's effect size, this is a considerable effect [83]. The beta weights, represented in table 8, suggested that the intra-cluster structure contributed more to the prediction of average update. The tolerances in the same table indicate a low collinearity (Tolerance > 0.5). This is due to the "orthogonality" of the two extracted factors.

6.2.2 Theoretical Results

The goal of our simulation is to show that OCTOPUS network will converge to a consensus given an appropriate mapping function. The role of intra and inter links on the convergence process is another aim of this simulation. The final objective is to determine the impact of intra and inter cluster synchronization on the overall network convergence.

For this purpose, we selected the saturation function $f(x)=\frac{2}{1+e^{\left(\frac{n-x}{an}\right)}}$ given in equation 26 as our mapping function. The starting point is to explore its behavior in reaching a convergence and stability point. Figure 22 indicates that all nodes will reach the saturation state. As the exchange of target list occurs between nodes, each node will update its target scope. The update stops when all targets are visible to a node. Figure 23 shows that using equation 40 for 4000 iterations, the increase of the number of nodes generates a higher negative Lyapunov exponent. A higher negative value is a sign of greater system stability. Figure 23 reveals also, that the mapping function used is conducive to a stability point for the communication system. It is also important to mention that the theoretical ratio $\frac{\lambda_2}{\lambda_n}$ (synch-ratio) obtained from equation 41 defines a lower bound for our communication system stability region and informs about the shape and behavior of the synchronization curve. Our system will synchronize and stabilize as long as its calculated ratio stays above the theoretical bound and follows its shape. The synch-ratio will be useful when exploring the impact of intra and inter links on the synchronization.

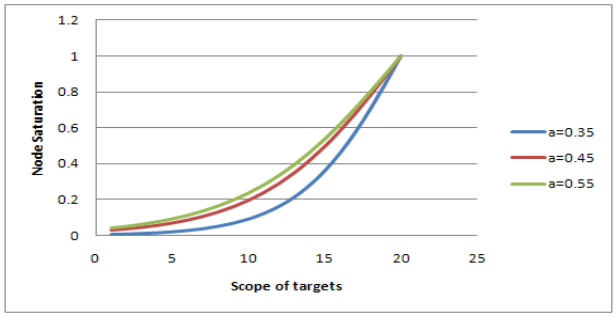

Figure 22: represents the evolution of a node's saturation state when the scope of targets increases.

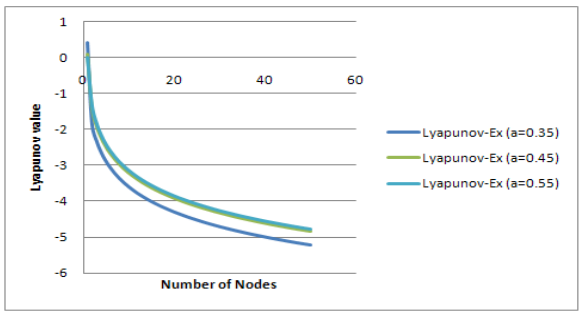

Figure 23: represents Lyapunov exponent values for different number of nodes and different mapping.

In general, by saturating the area with a higher number of nodes, the number of clusters tends to get smaller. More precisely, increasing the number of nodes leads to an increase, followed by a decrease of the number of clusters. This is illustrated in Figure 10, where twenty

nodes form one cluster. Moreover, the clustering technique is designed to favor dense intra cluster links over sparse inter cluster links when the number of nodes increases. Figure 12 shows that the ratio of intra cluster links to inter cluster links increases with an increase of number of nodes. This result forms the basis in the study of the impact of intra and inters cluster links in the convergence process.

Series of communication topologies were generated by increasing the number of nodes. For each topology, a number of measures were evaluated: The ratio of intra cluster links over the total number of links, intra clusters synchronization ratio (equation 33), inter clusters synchronization ratio (equation 35), and the overall system synchronization ratio.

In Figure 24, the calculated topology and intra cluster synchronization ratios are above the theoretical boundary and follow its trend. This confirms that the generated topologies will converge to a stable point. As the number of node increases, the system synchronization is dictated by intra cluster synchronization. The intra cluster convergence curve merge with the overall system synchronization curve.

In contrast to intra cluster synchronization, Figure 25 shows that smaller inter cluster synchronization ratios are obtained for smaller number of inter cluster links. In other words, a better synchronization is obtained when less inter cluster links are present.

A better visualization of the impact of intra and inter cluster links on the convergence of a topology is illustrated in Figure 26. Better system synchronization (low value) is obtained when the ratio of intra links is higher. Bad synchronization (high value) is obtained when the ratio of inter links is higher. Similarly, in Figure 27, a better system synchronization is obtained when intra cluster synchronization is dominant over inter cluster synchronization. Low system synchronization ratios are obtained for low intra cluster synchronization ratios.

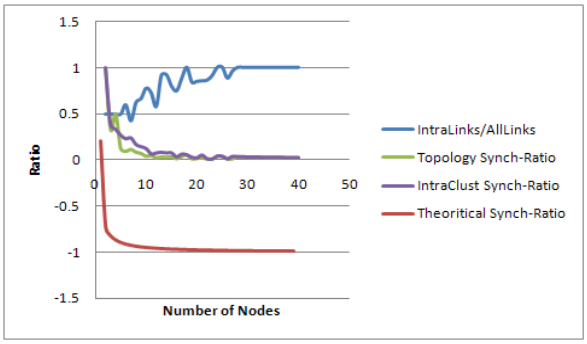

Figure 24: Network and intra cluster synchronization ratio when the number of nodes increases.

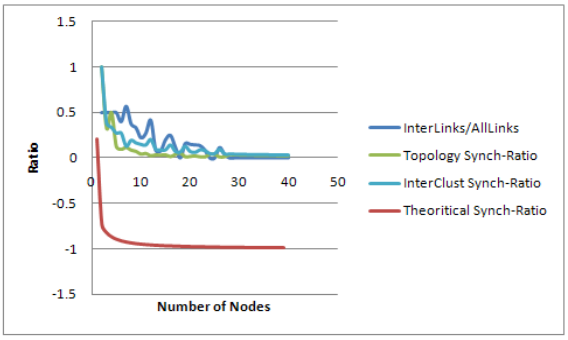

Figure 25 : Network and inter cluster synchronization ratio when the number of nodes increases.

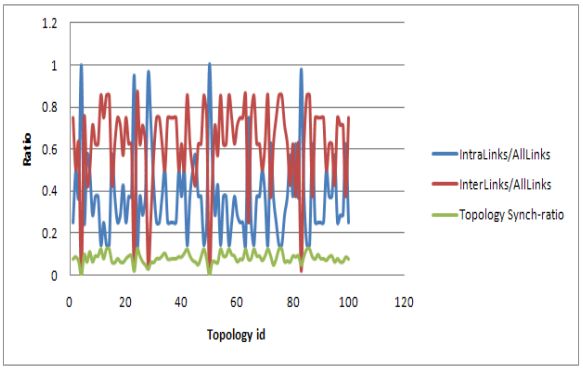

Figure 26: Variation of topology synchronization rate when intra and inter cluster links vary.

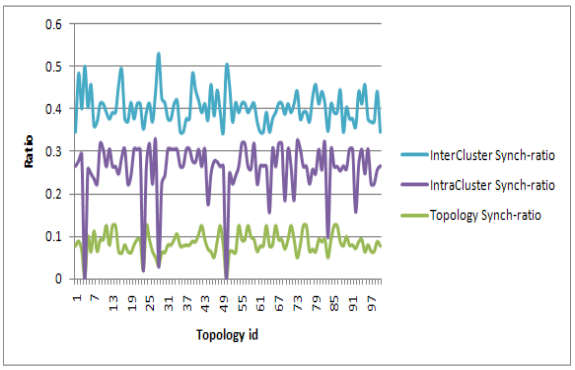

Figure 27: Variation of topology synchronization rate when intra and inter cluster synchronization rates vary.

6.3 Task Assignment

6.3.1 Assumptions

Few assumptions need to be clarified before presenting the simulation results. The techniques assume that the autonomous robots are identical and have equal initial energy. The robots and targets are, also, randomly dispersed in a geographical area. For simplification, a target or a task is non divisible and requires one robot to service it. In the explorative simulation, the selected robots are moved to the assigned targets in a way similar to equation 43. The robot's cost is measured by the distance from the robot initial position to the target position. The overall cost is a summation of all distances traveled by the robots. The workload distribution is measured by the number of targets assigned to a robot.

In the theoretical simulation, the deterministic factor for a robot selection in equation 57 is the cost of assignment. The winning node is the robot with the lowest cost. In equation 51, the selection is dictated by the value of x_{ij} in the output. 0 means the robots sits idle and 1 means it is active. The cost is evaluated by the dynamics of the differential equations 51 and 57. The workload distribution is measured by a fairness index.

6.3.2 Explorative Simulation

The objective of this section is to visualize, explore and compare the behavior of the three techniques before verifying it theoretically. The experiments focus on the fairness in workload distribution, the cost generated by the assignment and the execution time. Series of tests were devised to compare the three methods using MATLAB.

The first test was designed to visualize the behavior of each technique using three configurations:

- Number of robots equals the number of targets. (config. (a) → 4 robots and 4 targets).
- Number of robots is greater than the number of targets. (config. (b) → 4 robots and 2 targets).
- Number of robots is less than the number of targets. (config. (c) → 3 robots and 7 targets).

Applying these techniques on each configuration led to few observations. The choice of an assignment technique is sometimes irrelevant. Config. (b) in Figure 28, 29 and 30 shows the three techniques exhibiting similar behavior. Targets *T1* and *T2* are always serviced by the same robots, respectively *R1* and *R4*. This suggests the existence of configurations and scenarios where other factors besides the technique need to be considered.

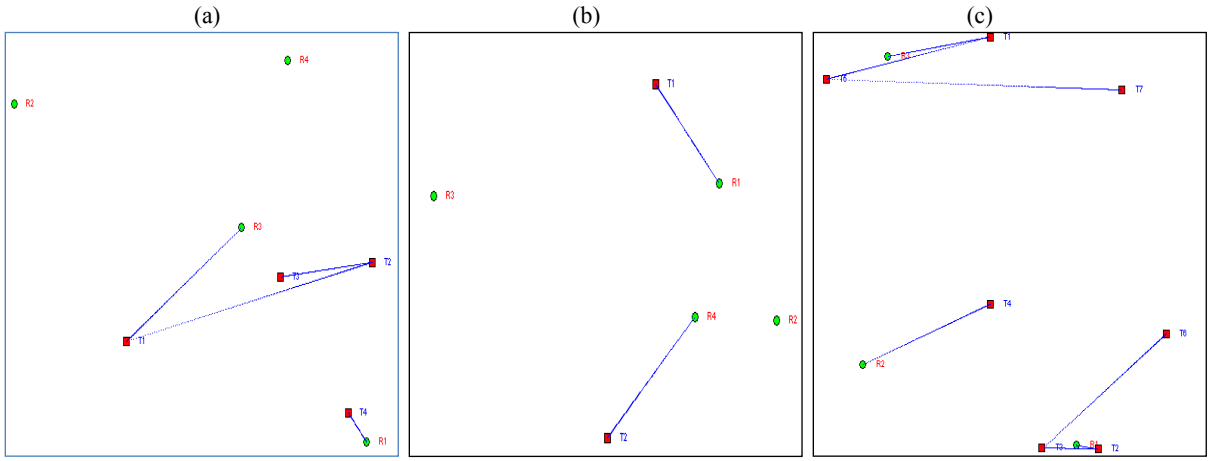

Figure 28: Target assignment using SOM for config (a), (b) and (c). *Note* : Circles denote robots (R) and squares denote targets (T).

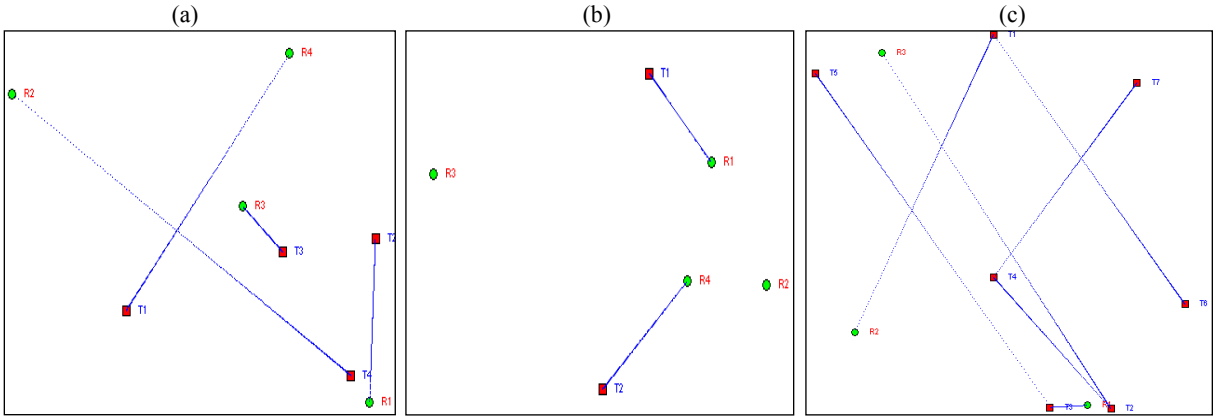

Figure 29: Target assignment using Hungarian method for config (a), (b) and (c).

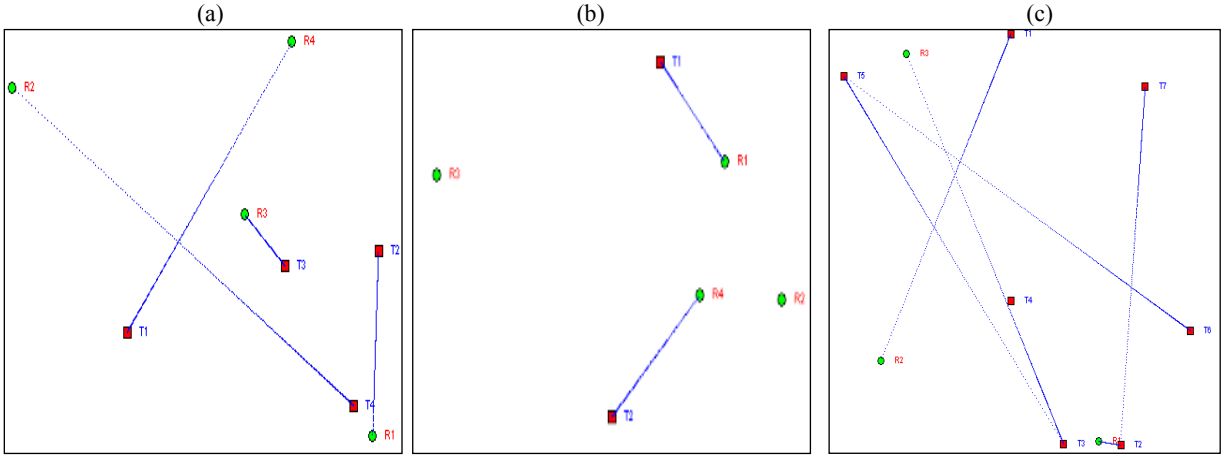

Figure 30: Target assignment using ILP method for config (a), (b) and (c).

The Hungarian and the ILP methods are apparently similar in behavior in most cases. Config. (a) and (b) in all figures show the targets being serviced by the same robots. The three methods, however, can yield three different solutions. Config. (c) in all figures present three different robot-to-target assignments. This implies the presence of dissimilarities in the three techniques.

Another characteristic exhibited by config. (a) is fairness. SOM focuses on cost since only two robots are selected to service four targets. The Hungarian and ILP methods, on the other hand, focus on load balancing. Each target is assigned a robot.

The second goal is to compare the cost generated by each method. For this purpose the number of robots was kept constant (4) and the number of targets was increased from 4 to 16. The result in Figure 31 shows that ILP method generates a higher cost compared to the Hungarian and SOM methods. The Hungarian cost shows an almost perfect monotonous increasing function. ILP and SOM, on the other hand, exhibit an almost perfect curve resemblance and tend to generate lower costs when the number of targets equal 5, 12 and15.

The third purpose is to compare the workload distribution generated by each method. Keeping the same configuration as the previous test, we evaluated the average workload for each robot under each method. Figure 32 shows that ILP and Hungarian balance the workload among robots whereas SOM approach favors some robots over others. This explains why the cost of ILP and Hungarian is slightly higher compared to SOM. Figure 32 shows, also, a small difference in robots workload using the ILP and Hungarian method. Robots 1, 2 and 3, experience different workloads under the two methods.

Another test conducted on the described methods is the evaluation of their running time. Many parameters can influence the accuracy of the results. These factors can range from the developer's skills to the host machine configuration and MATLAB capabilities. In this test, we limited ourselves to verifying the accuracy of the time complexity stated for each method.

For this purpose, we selected a configuration that includes 9 targets and measured the execution time for each method when the number of robots servicing the targets is increased from 1 to 4. Figure 36 shows that, for all techniques, the running time increases when the number of workers increases. A second observation is that ILP running time is always higher than Hungarian and SOM running time (factorial higher than polynomial). Third, SOM has a slightly lower execution time than Hungarian ($\theta(n^2) < \theta(n^3)$).

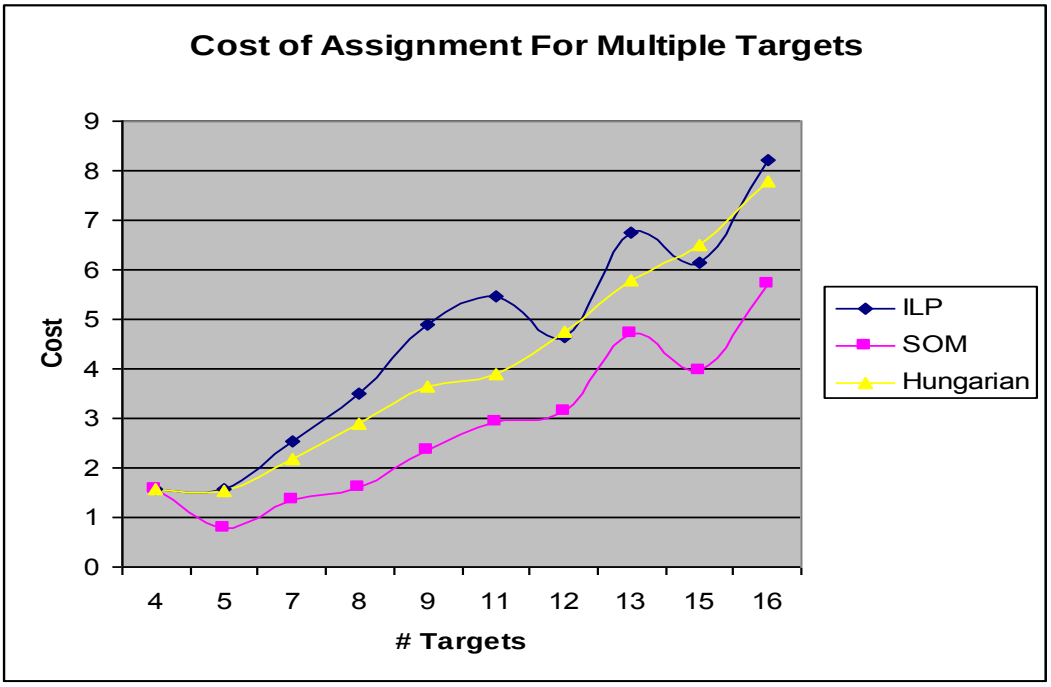

Figure 31: Cost generated by each method.

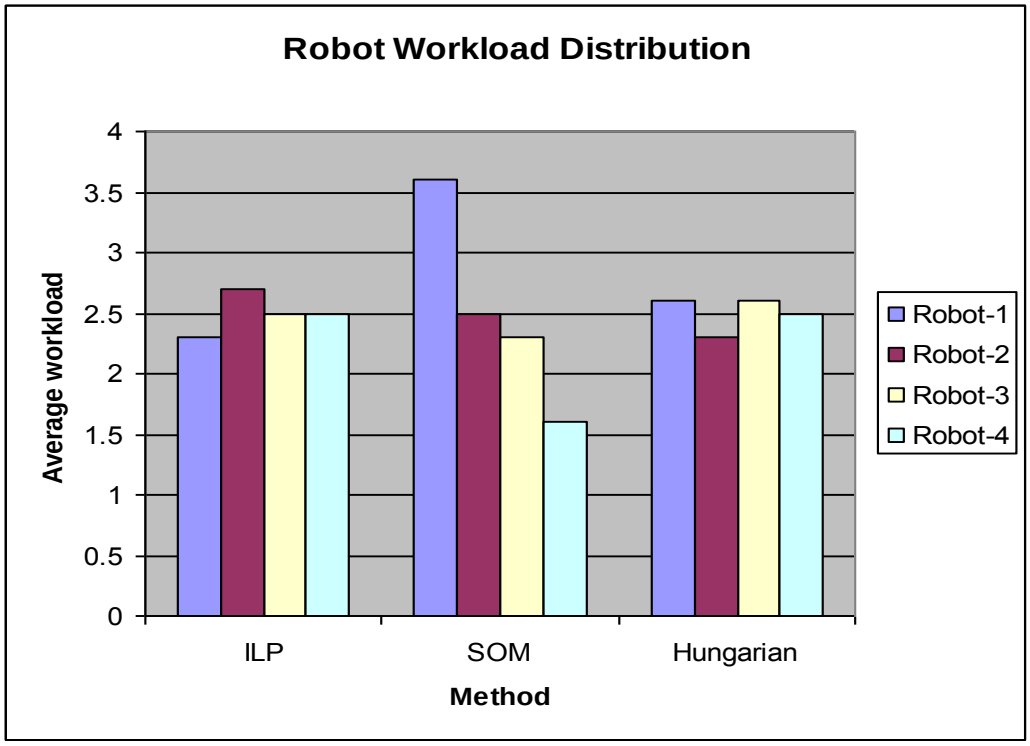

Figure 32: Average workload distribution generated by each method.

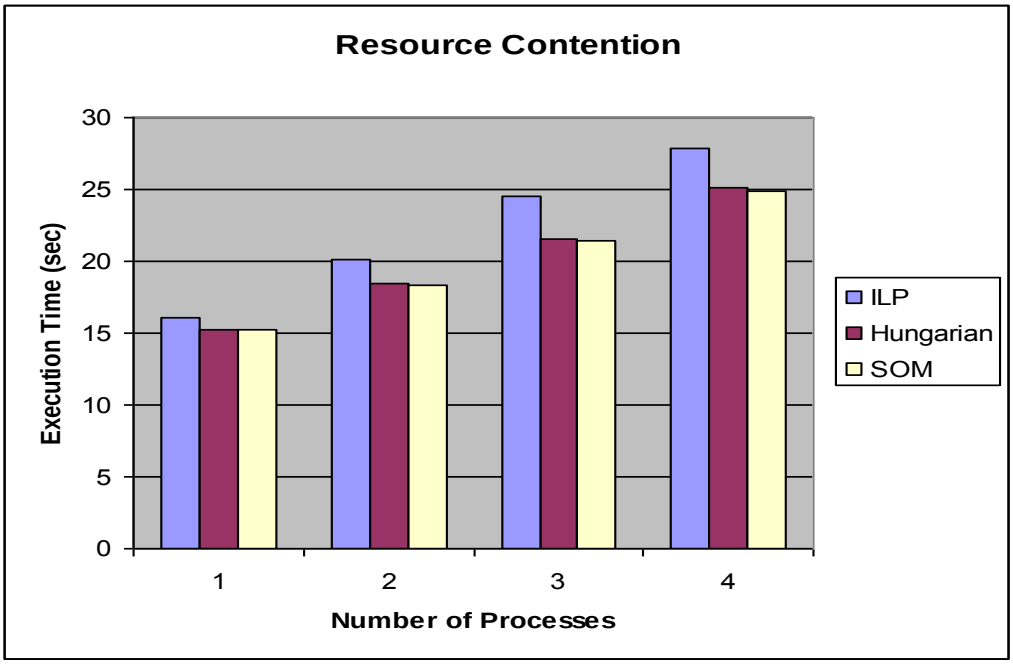

Figure 33: Task execution time for the three techniques when the number of processes increases.

In summary, this preliminary simulation revealed two major observations. First, ILP and the Hungarian method exhibit similar behavior in task assignment. Second, SOM focuses on cost whereas IPL and the Hungarian techniques focus on workload balancing.

6.3.3 Theoretical Results

The goal of this section is to reproduce and validate the observations obtained in the previous section using the differential equations developed for SOM and ILP techniques. The simulation is performed under MATLAB and uses similar robots and target configurations for its validation phase. As an example, we use configuration (a) in Figure 28 and 30 to compare the behavior of these two methods. Configuration (a) consists of four robots and four targets randomly spread in a limited area.

The reason for selecting this example is to validate and facilitate the interpretation of results. For SOM behavior, we applied Euler method on equation 57 and captured the dynamics of each robot's ODE when presented by an input (target). Figure 34a shows that R3 ODE converges rapidly to a stability point with a lower cost. This suggests that R3 is more appropriate to service T1. Similarly, Figure 34b, 34c and 34d show that R3, R3 and R1are good candidates to service respectively T2, T3 and T4. These results are similar to the one obtained by SOM previously.

The first step in simulating equation 51 is the selection of appropriate values for α and μ parameters. This step was accomplished by trial and error. The experimentation suggested that higher values of μ shortens the execution time and smaller values of α influence the weight and accuracy of the penalty function. Figure 35 shows that applying equation 51 on the list of targets resulted in the selection of four robots. This suggests that each robot will service one target. For

clarity only the dynamics of the active robots is presented. The result is similar to the one in the explorative simulation. Validating equations 51 and 57 was performed over few other configurations. The results were in general similar.

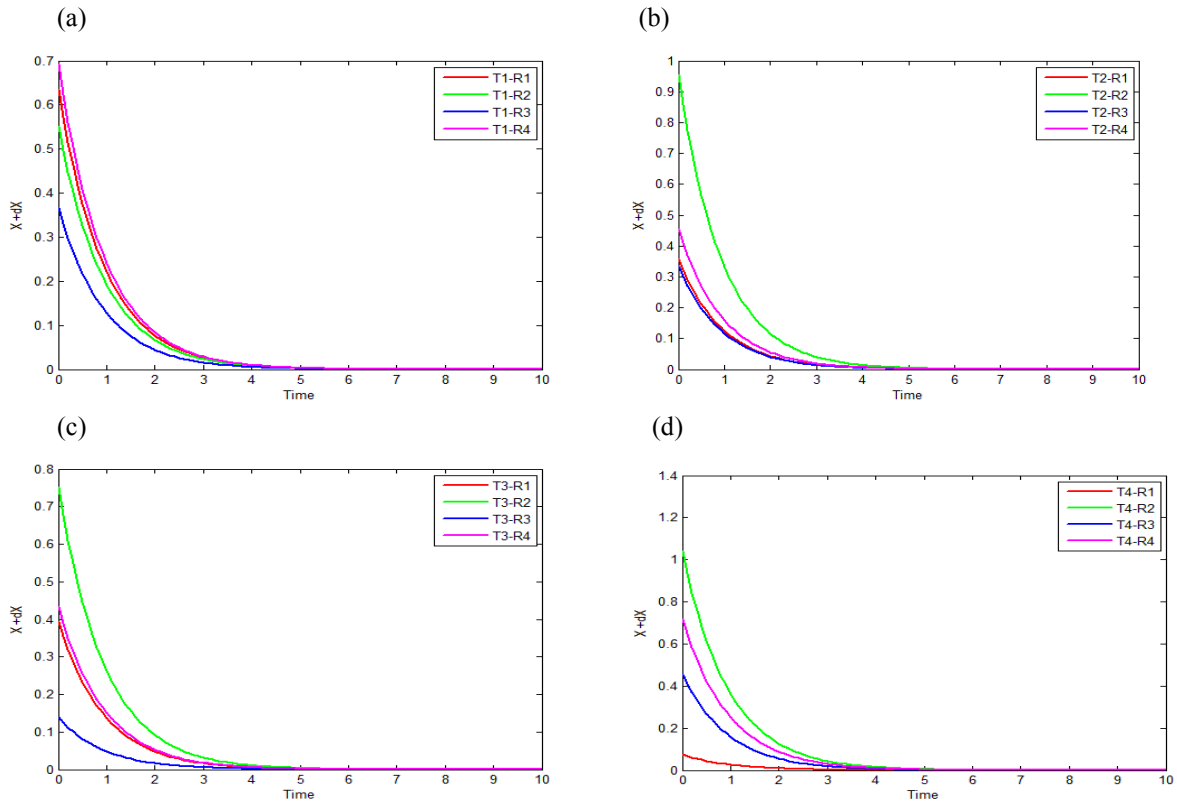

Figure 34: Robots ODE dynamics when targets are introduced (a)=T1, (b)=T2, (c)=T3 and (d)=T4.

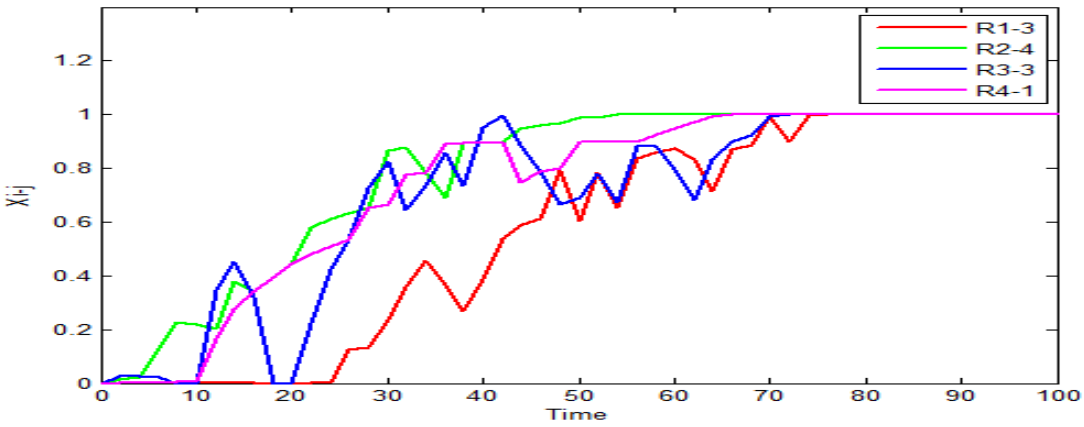

Figure 35: Robots selection for target servicing, (1=active, 0=idle).

The two properties, we intend to compare is the cost of assignment and the fairness in workload distribution of equations 51 and 57. Few random topologies (1000) of 10 robots and 10 targets were generated. We measured the total cost of assigning the robots and the fairness ratio for equations 51 and 57. Figure 36 confirms the results obtained in the previous simulation. ILP cost of assignment is most of the time higher than SOM's. Figure 37 shows that the fairness ratio of ILP is always higher than SOM's. This means that ILP balances the workload among robots compared to SOM.

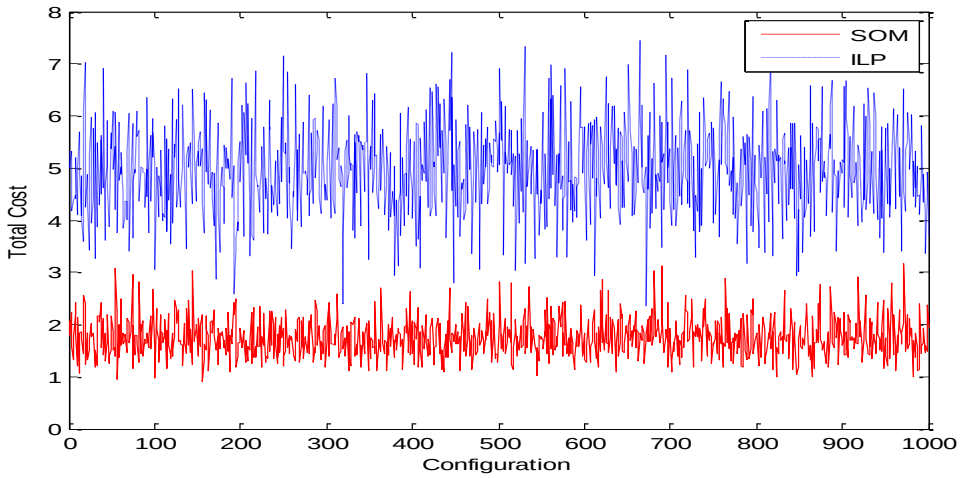

Figure 36: Cost of assignment for SOM and ILP.

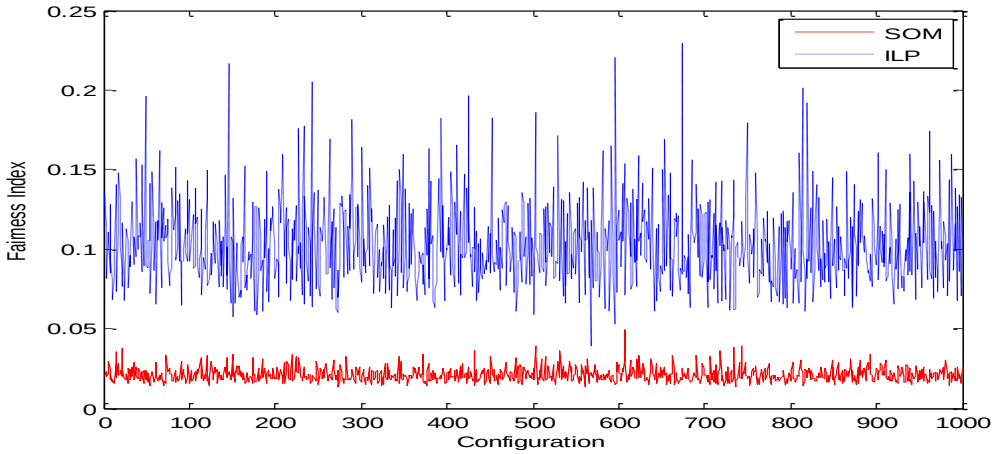

Figure 37: SOM and ILP fairness in workload distribution.

Another experiment was conducted to verify the impact of topology changes on the cost and workload distribution. The number of robots was kept constant (10) and the number of targets was increased, gradually, to 100. Figure 38 and 39 show that the behavior of the two equations was not altered by the changes. However the fairness decreased for both techniques whereas ILP cost increased considerably. This suggests that there are configurations where the cost will be a determinant factor and the fairness could be ignored.

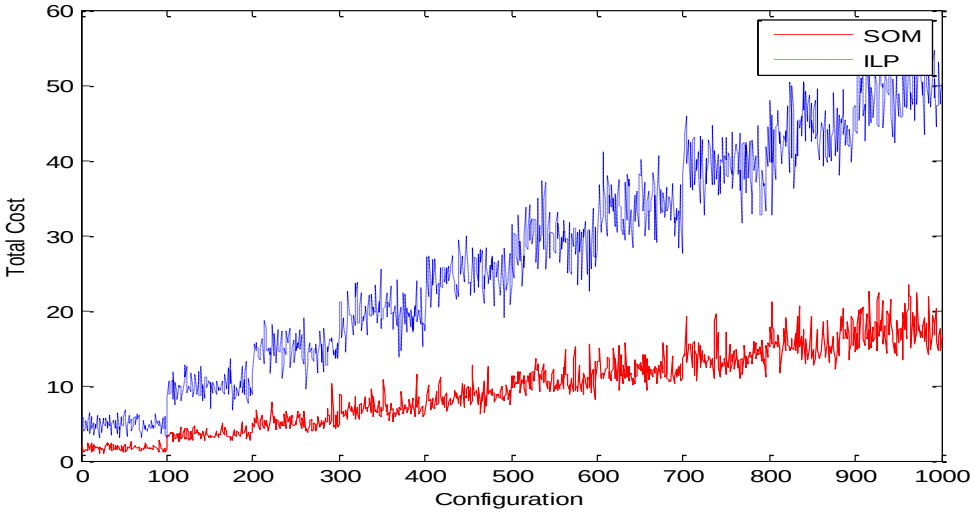

Figure 38: Cost of assignment when the number of target increases.

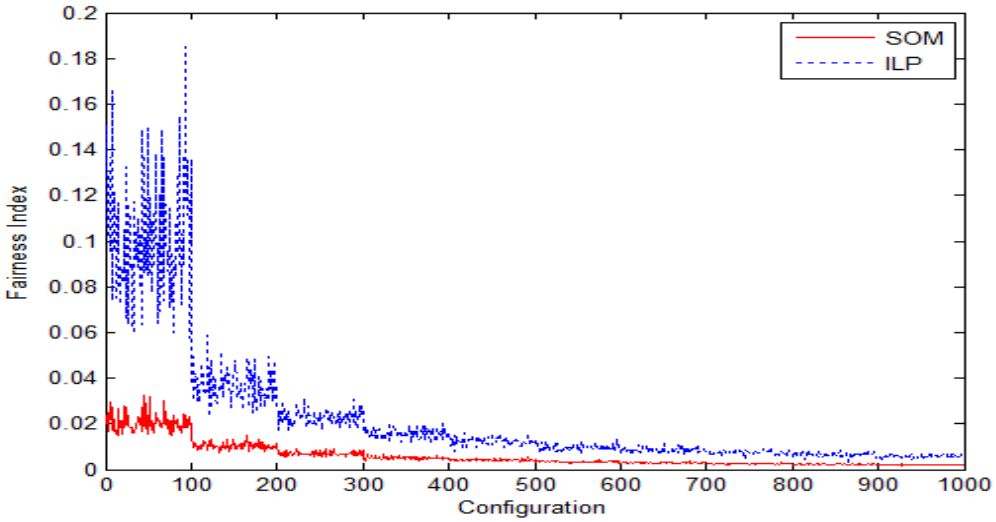

Figure 39: SOM and ILP fairness in workload distribution when the number of targets increases.

CHAPTER 7

CONCLUSIONS

This dissertation proposes, in a first phase, a distributed technique to build a communication topology for mobile sensor networks. The approach starts with clustering the nodes into separate groups. Communication among nodes of the same cluster is facilitated by the elected cluster-head and communication with other clusters is performed by the selected gateway node. The result of this technique is a communication map that specifies for each node its role in the network. The technique has many applications. As an on-demand, a node can request a re-configuration in case a failure occurs at the cluster-head or gateway level. Similarly, a node joining the network could request a role through a re-configuration. The approach can be employed as a scheduled technique for re-organization and re-configuration and an alternative to topology control and maintenance. However, an issue to consider is the cost of frequent updates and re-evaluation of the topology

In a second phase, the study investigates the effect of few topology structural properties on the consensus building among robots. Six metrics, algebraic connectivity ratio, average path, average cluster coefficient, average matching index, modularity and average participation, were investigated using a factor analysis technique. Only algebraic connectivity ratio, average matching index, modularity and average participation proved to have a major impact on the consensus building. The regression analysis proved that up to 50% in consensus variability could be explained by these four parameters. It also showed that the consensus building could be predicted and that the metrics describing the intra-cluster connection have more weight in the model.

In this study, only six parameters were investigated resulting in a linear model and 50% explained variance. This result could be improved if more structural and non-structural metrics were incorporated. Furthermore, a model describing the consensus building for any given topology with a random number of robots could be devised.

In a third phase, we compared three techniques for task allocation in a multi-agent system. SOM competitive approach uses the Euclidian distance to assign a target to the winning node. The Hungarian algorithm, searches, within a cost matrix, for a minimal set of zeros that appears exactly once in each row and column in order to solve the assignment problem. Finally, the linear programming method tries to minimize an objective function under certain constraints. The experiments show that, overall, the techniques generate different assignment solutions along with different costs and workload distribution; but for certain configurations the solutions are either identical or similar. The key factor in deciding which technique to select involves finding a trade-off between the cost of assignment, workload distribution and execution time. SOM offers the lowest cost and running time with the worst workload distribution. ILP generates a higher cost, a higher execution time and a balanced workload. Hungarian exhibits an in-between cost and acceptable running time while preserving a balanced workload. Thus suggesting the utilization of the Hungarian method is more appropriate for most scenarios. However in the case where robots have abundant power and the execution time is a concern, the selection of SOM is more suitable.

This comparison study assumes that all robots are identical and have equal initial energy. It also assumes that tasks are non-divisible and require only one robot. In real scenarios and most of the time this is not the case. A heterogeneous park of robots or divisible tasks will require the

definition of additional constraints and a new cost matrix that combines all parameters for the three methods.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. D. J. Baker and A. Ephremides, "A Distributed Algorithm for Organizing Mobile Radio Telecommunication Network," *in Proc. 2nd Int. conf. Distrib. Comput. Syst*., Apr. 1981, pp. 476-483.

2. M. Gerla and J.T.C. Tsai, "Multicluster, Mobile, Multimedia Radio Network," *Wireless Networks*, Vol. 1, pp. 255-265, 1995

3. S. Basagni, "Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks," *in Proc. Veh. Technol. Conf.*, Sept. 1999, vol.2, pp. 889-893.

4. H. Tan, W. Zeng and L. Bao, "PATM: Priority-Based Adaptive Topology Management for Efficient Routing in Ad Hoc Networks," in *Proc. Int. Conf. Comput. Sci*., 2005, pp. 485-492.

5. W.R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *Proc. 33rd Hawaii Int. Conf. Syst. Sci.*, 2000, vol. 8, pp. 8020.

6. F. Comellas and S. Gago, "Synchronizability of Complex Networks," *J. Phys. A: Math. Theor*., vol. 40, no. 17, pp. 4483-4492, Apr. 2007.

7. F.M. Atay, T. Biyikoglu and J. Jost, "Network Synchronization: Spectral Versus Statistical properties," *Physica*, vol. 224, no. 1–2, pp. 35–41, 2006.

8. J. Z. Yang and M. Zhang, "Chaos Synchronization in Complex Networks," Chin. Phys. Lett., vol. 22, no. 9, pp. 2183-2186, Sep. 2005.

9. P. Hovareshti and J. S. Baras, "Consensus Problems on Small World Graphs: A Structural Study," in *Proc. Int. Conf. Complex Syst.*, May 2006.

10. S. Kar, J.M.F. Moura, "Topology for Global Average Consensus," in *Proc. 40th Asilomar Conf. Signals, Syst. Comput.*, Oct. 2006, pp. 276-280.

11. P. V. Sander, Denis Peleshchuk and Barbara J. Grosz, "A Scalable, Distributed Algorithm for Efficient Task Allocation**,"** in *Proc. 1st Int. Joint Conf. Auton. Agents Multiagent Syst*., Bologna, Italy, Jul. 2002, pp. 1191-1198.

12. P. B. Sujit, A. Sinha and D. Ghose, "Multiple UAV Task Allocation using Negotiation," in *Proc. 5th Int. Joint Conf. Auton. Agents Multiagent Syst.*, May 2006, pp.471-478.

13. P.B. Sujit, A. Sinha, and D. Ghose, "Multi-UAV Task Allocation using Team Theory," in *Proc. IEEE Conf. Decision Control*, Seville, Spain, Dec. 2005, pp. 1497-1502.

14. Campbell and A. S. Wu, "Learning and Exploiting Knowledge in Multi-Agent Task Allocation Problems," in *Proc. Evol. Comput. Multiagent Syst. Simulation*, London, U.K., Jul. 2007, pp. 2637-2642.

15. M. J. Mataric, G.S. Sukhatme and E.H. Ostergaard, "Multi-Robots Task Allocation in Uncertain Environments," *Auton. Robots*, vol. 14, no. 1, pp. 255-263, 2003.

16. J.Butterfield, O. Chadwicke and B. Gerkey, "Multi-Robot Markov Random Fields," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.*, Estoril, Portugal, May 2008, pp. 1211-1214.

17. D. Goldberg, V. Cicirello, M. B. Dias, R. Simmons, S. Smith, and A. Stentz, "Task Allocation using a Distributed Market-Based Planning Mechanism," in *Proc. 2nd Int. Joint Conf. Auton. Agents Multiagent Syst.,* Melbourne, Australia Jul. 2003, pp. 996-997.

18. K. Manousakis, L. Kant, K. Young, C. Graff, M. Patel, and D. Yee, "Wireless Test-bed Implementation of Cognitive Topology Control for Power Efficient Connected Networks," *Mil. Com. Conf.*, pp. 1090-1095, Nov 2011.

19. F. T. Lin, C.Y. Lee and C. S. Yang, "A Distributed Reliable and Energy-Efficient Topology Control Algorithm in Wireless Sensor Network," *Int. Conf. Info. Sci. Appl.*, pp. 1-6, May 2012.

20. J. Ma, M. Gao, Q. Zhang and M. L. Ni, "Energy-Efficient Localized Topology Control Algorithms," *IEEE Trans. Parallel Distrib. Syst*., vol. 18, pp. 711-720, May 2007.

21. C. Xiuzhen, M. Ding, D. H. Du and X. Jia, "Virtual Backbone Construction in Multi-Hop Ad Hoc Wireless Networks," *Wireless Commun. Mobile Comput.*, vol. 6, pp. 183-190, March 2006.

22. V. Gayathri, E. Sabu and T. Srikanthan, "Size-restricted Cluster Formation and Cluster Maintenance Technique for Mobile Ad Hoc Networks," *Int. J. Netw. Manag.*, vol. 17, pp. 171-194, March 2007.

23. P. Tsigas, "A Self-stabilizing (K-R)-Clustering Algorithm with Multiple Paths for Wireless Ad-hoc Networks," *Dist. Comp. Sys.*, pp. 353 - 362, Jun. 2012.

24. F. G. Nocetti, J. S. Gonzales and I. Stojmenovic, "Connectivity Based K-hop Clustering in Wireless Networks," in *Proc. 35th Annu. Hawaii Int. Conf. Syst. Sci.*, 2002, vol. 7, pp. 188.

25. J. H. Li, M. Yu, and R. Levy, "Distributed Efficient Clustering Approach for Ad Hoc and Sensor Networks," in *Proc. 1st Int. Conf. Mobile Ad hoc and Sensor Netw*., Dec. 2005.

26. P. Wan, K. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," in *Proc. INFOCOM*, Apr. 2004, vol. 9, pp. 141-149.

27. T. T. Wu and K. F. Su, "Determining Active Sensor Nodes for Complete Coverage without Location Information," *Int. J. Ad Hoc Ubiquitous Comput*., vol. 1, pp. 38-46, Nov. 2005.

28. T. W. Haynes, S. T. Hedetniemi and P.J. Slater, *Fundamentals of Domination in Graphs.* New York: Marcel Dekker, 1998.

29. L. V. Fausett, *Fundamentals of Neural Networks Architectures: Algorithms and Application.* New Delhi, India: Pearson Education, 2005.

30. L. Ruan, D.H. Du, X. Jia, W. Wu, Y. Li and K.-I Ko, "A Greedy Approximation for Minimum Connected Dominating Sets," *Theoret. Comput. Sci.*, vol. 339, pp. 325-230, Dec. 2004.

31. M. Min, C.X. Huang, S. C.-H. Huang, W. Wu, H. Du, and X. Jia, "Improving Construction of Connected Dominating Set with Steiner Tree," *Wireless Sensor Netw.*, vol. 35, pp. 111-119, May 2006.

32. K.M. Alzoubi, P.-J. Wan and O. Frieder, "Distributed Heuristics for Connected Dominating Sets in Wireless Ad Hoc Networks," *J. Commun. Netw.*, vol. 4, pp. 24-29, Mar. 2002.

33. X. Cheng, X. Huang, D. Li, W. Wu and D.-Z. Du, "Polynomial-Time Approximation Scheme for Minimum Connected Dominating Set in Ad Hoc Wireless Networks," *Networks,* vol. 42, pp. 202-208, Sep. 2003.

34. C. H. Papadimitriou, "On the Complexity of Integer Programming," *J. ACM,* vol. 28, pp.765-768, Oct. 1981.

35. T.W. Haynes, S. T. Hedetniemi and P.J. Slater, *Domination in Graphs Advanced Topics.* New York: Marcel Dekker, 1998.

36. M. Chatterjee, S.K. Das and D. Turgut, D., “WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks,” *Cluster Comput. J.*, vol.5, no 2, pp. 193-204, Apr. 2002.

37. H. Lui and R. Gupta, “Selective Backbone Construction for Topology Control,” in *Proc. 1st IEEE Int. Conf. Mobile Ad-hoc and Sensor Syst. (MASS)*, Oct. 2004, pp. 41-50.

38. L. A. Wolsey, *Integer Programming*. New York: Wiley-Interscience, Sep. 1998.

39. G. Sierksma, G. A. Tijssen and P. V. Dam, Linear and Integer Programming: Theory and Practice. New York: Marcel Dekker, Mar. 1996.

40. B. H. Junker and F. Schreiber, *Analysis of Biological Networks*. New York: Wiley, Mar. 2008.

41. T. Hatanaka and M. Fujita, "Dynamic Topology Optimization for Dependable Sensor Networks," *Int. Conf. Contr. Appl.*, pp. 274-279, Sep. 2010.

42. Y. Hu, "The Statistical Characteristics of Internet Topology Models," *Int. Conf. Inf. Sci. Eng.*, pp. 2475 - 2478, Dec. 2010.

43. J. McNair, J. Antoon, A. Gordon-Ross, K. Cason and N. Fitz-Coy, "Topology Design and Performance Analysis for Networked Earth Observing Small Satellites," *Mil. Com. Conf.*, pp. 1940-1945, Nov. 2011.

44. A. Jamakovic, S. Uhlig, “On the Relationship between the Algebraic Connectivity and Graph’s Robustness to Node and Link Failures,” in *Proc. 3rd EuroNGI Conf. Next Generation Internet Networks*, May 2007, pp. 96-102.

45. M. E. J. Newman and M.Girvan, “Finding and Evaluating Community Structure in Networks,” *Phys. Rev. E*, vol. 69, no. 2, pp. 1–15, 2004.

46. E. Ziv, M. Middendorf, and C. H. Wiggins, “Information-Theoretic Approach to Network Modularity,” *Phys. Rev. E*, vol. 71, no. 4, p. e046117, Apr. 2005.

47. E. Ben-Naim, H. Frauenfelder, Z. Toroczkai, *Complex Networks*. New York: Springer, Sep. 2004.

48. Reka and A.L. Barabasi, “Statistical Mechanics of Complex Networks,” *Rev. Modern Phys.*, vol. 74, no. 1, pp. 47–97, Jan. 2002.

49. H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker and W. Willinger, “Network Topology Generators: Degree-Based vs. Structural,” in *Proc. ACM SIGCOMM*, 2002, pp. 147-150.

50. S. Boccalettia, V. Latora, Y. Moreno, M. Chavez and D.U. Hwanga, “Complex Networks: Structure and Dynamics*,” Phys. Rep.*, pp. 175–308, Feb. 2006.

51. M. E. J. Newman, “The Structure and Function of Complex Networks,” *SIAM Rev.*, vol. 45, pp. 167–256, 2002.

52. J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*. Berlin, Germany: Springer-Verlag, 2001.

53. L. A. N. Amaral, A. Scala, M. Barthelemy and H. E. Stanley, “Classes of Small-World Networks,” in *Proc. Nat. Acad. Sci.*, Jul. 2000, vol. 97, no. 2, pp. 11149–52.

54. R. Reyment and K.G. Joreskog, *Applied Factor Analysis in the Natural Science*. Cambridge, U.K.: Cambridge Univ. Press, 1993.

55. A. L. Comrey and H.B. Lee, *A First Course in Factor Analysis*. Mahwah, NJ: Lawrence Erlbaum, 1992.

56. A.B. Costello and J.W. Osborne, "Best Practices in Exploratory Factor Analysis: Four Recommendations for getting the most from your Analysis," *Res. Evaluation*, vol. 10, no. 7, pp. 1–9, Jul. 2005.

57. D. F. Polit and C. T. Beck, *Nursing Research: Principles and Methods*. Baltimore, MD: Williams & Wilkins, Jun. 2000.

58. R. L. Gorsuch, *Factor analysis*. Mahwah, NJ: Lawrence Erlbaum, Nov. 1983.

59. D. L. Rubinfeld, "Reference Guide on Multiple Regression," Reference manual on Scientific Evidence, 2000.

60. N. L. Leech, K. C. Barrett and G. A. Morgan, *SPSS for Intermediate Statistics use and interpretation*. Mahwah, NJ: Laurence Erlbaum, Jul. 2007.

61. K. Kaneko, *Theory and applications of coupled map lattices*. New York: Wiley, 1993.

62. T. Kato, *Perturbation Theory for Linear Operators*. New York: Springer-Verlag, 1976.

63. O. Edwards, *Chaos in Dynamical Systems*. Cambridge, U.K.: Cambridge Univ. Press, 1993.

64. N. Hiroyuki, *Introduction to Chaos*. Philadelphia, PA: Institute of Physics, 1999.

65. K. Kaneko and I. Tsuda, *Complex Systems: Chaos and Beyond*. New York: Springer, 2001.

66. S. S. Haykin, *Neural Networks: A Comprehensive Foundation*. 2nd ed., Englewood Cliffs: Prentice Hall, Jul. 1998.

67. K. Obermayer and T.J. Sejnowski, *Self-organizing Map Formation: Foundations of Neural Computation*. 1st ed., Cambridge, MA: MIT Press, 2001.

68. T. kohonen, *Self-Organizing Map*. 3rd ed., New York: Springer-Verlag, 2001.

69. H. N. Gabow, "Data Structure for Weighted Matching and Nearest Common Ancestors with Linking," in *1st Annu. ACM Symp. Discrete Algorithms*, 1990, pp. 434-443.

70. D. Jungnickel, *Graph Networks and Algorithms: Algorithms and Computation in Mathematics*. 3rd ed., New York: Springer-Verlag, 2008.

71. F. Andreas, "On Kuhn's Hungarian Method –A Tribute from Hungary," Egerváry Research Group, Budapest, Hungary, October 2004.

72. L. Lovász and M.D. Plummer, *Matching Theory*. 1st ed., Amsterdam, Netherlands: Elsevier, 1986.

73. L. R. Foulds, *Optimization Techniques, An Introduction*. New York: Springer, 1981.

74. M.S. Bazaraa, J.J. Jarvis, H.D. Sherali, *Linear Programming and Network Flows*. 2nd ed., New York: Wiley, Jan. 1990.

75. A. Cichocki, R. Unbehauen, *Neural Networks for Optimization and Signal Processing*. 1st ed., New York: Wiley, 1993.

76. D.G. Luenberger, *Linear and Nonlinear Programming*. 2nd Ed., Reading, MA: Addison-Wesley, 1984.

77. E. Hairer, S.P. Nørsett, G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*. 2nd ed., New York: Springer, 2008.

78. J.C. Butcher, *Numerical Methods for Ordinary Differential Equations*. 2nd ed., New York: Wiley, 2008.

79. T. Chen, H. Zhang, X. F. Zhou,G. M. Maggio, and I. Chlamtac, **"**CogMesh: A Cluster Based Cognitive Radio Mesh Network**,"** in *Proc. 2nd IEEE Int. Symp. New Frontiers Dynam. Spectrum Access Netw.*, Apr. 2007, pp. 168–178.

80. *Statistics Toolbox User's Guide*. Natick, MA: MathWorks Inc., 2009, MATLAB Tutorials, Version 7.

81. L. Barolli, H. Ando, F. Xhafa, A. Durresi, R. Miho and A. Koyama, "Evaluation of an Intelligent Fuzzy-based Cluster Head Selection System for WSNs using Different Parameters," *AINA workshops*, Mar. 2011.

82. Q. Zhang and D. Agrawal, "Dynamic Probabilistic Broadcasting in Manets," *J. Par. Dist. Comput.*, Feb. 2005.

83. J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Mahwah, NJ: Lawrence Erlbaum, 1988.