CT-759-SS1
Don.- 2014

Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara

Estrategias Locales para Formación y Mantenimiento de Redes Inalámbricas

Tesis que presenta:

J. Guadalupe Olascuaga Cabrera

para obtener el grado de:

Doctor en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Directores de Tesis

Dr. Luis Ernesto López Mellado
Dr. Andrés Méndez Vázquez

CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, Abril de 2013.

Estrategias Locales para Formación y Mantenimiento de Redes Inalámbricas

Tesis de Doctorado en Ciencias
Ingeniería Eléctrica

Por:
J. Guadalupe Olascuaga Cabrera
Maestro en Ciencias
CINVESTAV Unidad Guadalajara 2007-2009

Becario de Conacyt, expediente no. 212765

Directores de Tesis
Dr. Luis Ernesto López Mellado
Dr. Andrés Méndez Vázquez

CINVESTAV del IPN Unidad Guadalajara, Abril de 2013.

Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara

Localized Strategies for Wireless Network Formation and Maintenance

A thesis presented by:
J. Guadalupe Olascuaga Cabrera

to obtain the degree of:
Doctor of Science

in the subject of:
Electrical Engineering

Thesis Advisors:
Dr. Luis Ernesto López Mellado
Dr. Andrés Méndez Vásquez

CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, April 2013

Localized Strategies for Wireless Network Formation and Maintenance

Doctor of Science Thesis
In Electrical Engineering

By:
J. Guadalupe Olascuaga Cabrera
Master of Science
CINVESTAV Unidad Guadalajara 2007-2009

Scholarship granted by CONACYT, no. 212765

Thesis Advisors:
Dr. Luis Ernesto López Mellado
Dr. Andrés Méndez Vázquez

CINVESTAV del IPN Unidad Guadalajara, April 2013.

Estrategias Locales para Formación y Mantenimiento de Redes Inalámbricas

Resumen

El objetivo de esta tesis es la creación y mantenimiento de redes de dispositivos móviles. Aproximaciones multiagentes basados en estrategias de autoorganización son propuestas con el objetivo de construir estructuras virtuales, detección de segmentación y recuperación de la red. En esta aproximación cada dispositivo móvil inalámbrico es controlado por un agente multirole el cual realiza sus tareas de forma eficiente usando unicamente interacciones locales. La administración de los roles permite la reconfiguración de la estructura, cuando los nodos dejan o llegan a la red, dando como resultado un comportamiento emergente global complejo. El ahorro de la energía en los nodos se logra ajustando el tiempo de intervalo de transmisión y la potencia de transmisión después que la red se ha formado. Dos métodos globales fueron diseñados con el fin de comparar los resultados de las estrategias locales con resultados cercanos al óptimo. La primera aproximación global es basada en el árbol de expansión mínimo, al cual se le asignaron pesos en los enlaces de una forma estratégica para obtener la capacidad de cada nodo en la red. La segunda aproximación es un algoritmo de optimización basado en partículas que busca en el espacio de soluciones una estructura cercana a la óptima. Las simulaciones muestran el desempeño de los algoritmos respecto a el consumo de energía y reconfiguración.

Localized Strategies for Wireless Network Formation and Maintenance

Abstract

This thesis deals with creation and maintenance of mobile devices networks. Multi-agent approaches based on self-organization strategies are proposed for building a virtual backbone, and segmentation detection and recovery. In these approaches each mobile device is controlled by a multi-role agent, which performs these tasks efficiently through local interactions; role management allows backbone reconfiguration when the nodes leave or join the network yielding a complex global emergent behavior. Energy saving is achieved by adapting the time interval and transmission power after the network formation. Two global approaches are also designed for comparing the results of distributed algorithms against an optimal solution. The first global approach is based on the minimum spanning tree, which is strategically weighted to obtain the capacity of nodes in the network. The second one is a particle swarm optimization algorithm that searches in the space of solutions in order to find a structure close to optimal. Simulations show the algorithm performance regarding energy saving and segmentation recovery.

Agradecimientos

Agradezco en primer lugar a mi familia por su apoyo incondicional.

A mis asesores por su guía persistente.

A CONACYT por el apoyo económico.

Contents

Introduction

Wireless ad-Hoc network systems are becoming increasingly complex due to the size and heterogeneity of the underlying hardware, and the complex interactions among their elements. In recent years, many research efforts have focused on topology construction for wireless ad-hoc networks, which have many important applications including disaster recovery, military operations, forest vigilance, etc. These types of networks are set up on-the-fly by using a set of wireless devices; specifically, these networks have no fixed infrastructure, and consist of a dynamic collection of nodes.

Problem Definition

All nodes are functionally equal and each of them can act as a router and without a centralized control. Therefore, virtual backbone construction and topology control are widely used approaches to ease implementation of various problems such as routing and resource management strategies in ad-hoc wireless networks. Current challenging issues of wireless networks mainly deal with efficient networking and network maintaining techniques aiming to allow energy saving and handling node's mobility. Other important features desired for networking strategies are scalability, i.e., independence of the number of nodes.

Wireless ad-hoc networks require a special management because of their hardware and energy limitations. For example, they often are powered by batteries; they have limited memory size, and limited computational power, etc. Therefore, memory usage and energy conservation are critical issues in these kinds of networks.

An immediate implication of these limitations is the necessity of reducing communication in order to avoid a premature drop out of a node from the network under the connectivity constraint.

It is desirable to design algorithms that are able to operate in a distributed fashion; due to the fact that ad-hoc wireless networks are highly dynamic and autonomous. In addition, it is also desirable for these algorithms to implement an efficient use of its energy without jeopardizing communication between the nodes.

These are some of the reasons why new algorithms are needed. The basic characteristics these new algorithms should have are:

1. Adaptability to network topology changes, energy constraints, etc.
2. Robustness against failures.
3. Working in a localized and decentralized way.

Thesis Objective

In order to cope with the previous characteristics, we propose to draw inspiration from biological systems. These systems show an effective ability to adapt to constant and sudden changes in different environments. Most of these systems are composed of a larger number of dynamic, autonomous, and distributed entities. These entities generate effective adaptive behaviors through the use of local policies, i.e., self-organization, they cooperate with each other at local level due to global knowledge is not available within a single neighborhood of the system. This produces an emergent behavior that can be used to accomplish the global goals. This principle can be copied in ad-hoc networks. For example, a way to support efficient communication between nodes in a wireless network is to develop a wireless virtual backbone architecture, whose main goal is to broadcast a message throughout the network. In order to reach this goal is convenient to design a distributed and self-organizing algorithm featuring adaptability, robustness, and scalability.

In this specific case, several constraints regarding network connectivity and energy conservation have to be considered. For this, we are using self-organization strategies to connect a set of nodes by implementing a nature inspired cluster-based algorithm. This approach reduces both computation complexity, and energy consumption of the whole ad-hoc network, that is, the network lifetime can be prolonged. This clustering is done by grouping nodes inside a certain transmission area. Each of these nodes plays different roles namely leader, gateway, bridge and member; each of them is controlled

by a designated leader agent. This leader agent is selected according to a weight which is composed by the residual energy and the number of neighbors. In addition, some metrics are necessary to evaluate the reduction of the transmission power based in optimizing power of a node depending of its role.

Several techniques based on flooding strategies have been proposed to cope with these problems; three representative approaches reported in recent literature are: a) Connected Dominating Set (CDS) (18, 25). b) Multi-Point Relay (MPR) (32, 38, 43, 44), and c) cluster-oriented strategies (1, 14, 21, 26, 48, 49).

Common features of these approaches are the high computational complexity of the strategies requiring high energy consumption, and the lower capacity to maintain the connectedness of the network faced with node's mobility.

Main Contributions

The approach held in this thesis for coping with the above mentioned drawbacks are *self-organization-based* approaches (15, 30), whose principle is to use local rules to produce a global behavior: distributed local interactions between neighbor nodes in the network yield a virtual backbone.

Two different algorithms are described throughout this thesis. One is an improvement of the Multi-Agent Communication (MWAC) algorithm, the other one is a new strategy called Energy-Efficient Self-Organized Algorithm (EESOA) that outperforms the former. The proposed distributed algorithms are based on a multi-agent self-organization method; they are power-efficient and scalable for wireless ad-hoc and sensor networks. The algorithms are designed to operate in a localized way, reducing the complexity of interactions, and energy consumption.

The performance of these algorithms have been evaluated via simulation using the Network simulator NS-2; numerous experiments have demonstrated energy saving, segmentation detection ability, and network recovery capabilities in the whole network.

Finally, it is well known that a broadcasting based on Minimum Spanning Tree (MST) consumes energy within a constant factor of the optimum. That is why our results are compared against the one obtained using a global networking procedure based on a Prim's algorithm. This has been done in order to show that our algorithm is able to obtain a good approximation of a global solution.

Although a MST is a sparse connected subgraph, it is often not considered a good topology since close-by agents in the original graph G might end up far away. That is why a particle swarm optimization algorithm was designed in order to compare the proposed EESOA algorithm.

Structure of the Thesis

The remainder of this thesis work is organized as follows:

- Chapter 1 overviews different strategies for networking in wireless ad-hoc networks.
- Chapter 2 describes the self-organization strategies for building a virtual backbone in the network.
- Chapter 3 is devoted to design a global strategy to find an optimum virtual backbone by using the independent set theory.

  Chapter 4 shows the behavior and performance of the proposed algorithms based on NS-simulations.
- Chapter 5 presents some concluding remarks and sketches out future research work on the matter.

Chapter 1

Networking Approaches

Abstract. This chapter summarizes the problem of flooding the network, and surveys different strategies for building a structure in wireless ad-hoc networks, namely Multi-Point Relay, Connected Dominating Set and group-based strategies.

1.1 Network Flooding

Pure flooding is simple and easy to implement. This approach guarantees with a high probability, that each non isolated node will receive the broadcast packet. The main disadvantage of this technique is that it consumes large amounts of bandwidth because of many redundant retransmissions. Furthermore, not all the nodes need to retransmit the packet after receiving it.

Several algorithms have been proposed to optimize flooding such as MPR, CDS, and cluster-oriented strategies. We briefly review such approaches in the next sections.

1.2 MultiPoint Relay

The objective of MPR is to reduce the flooding of broadcast packets in the network by minimizing the duplicate retransmissions locally (31). Each node selects a subset of neighbors called MPRs to retransmit broadcast packets (Figure 1.1). This allows neighbors, which are not in the MPR set, to read the message without retransmitting it. This prevents the network flooding, i.e., the so-called broadcast storm (32)(43). Of

course, each node must select a MPR set among its neighbors such that it guarantees that all two-hop away nodes will get the packets, i.e., all two-hop away nodes must be a neighbor of a node in the MPR set. Several polynomial-time algorithms have been proposed to determine a MPRs set with minimal cardinality. However, it has been proven that the selection of a minimum size MPRs set is *NP-hard* (38)(44).

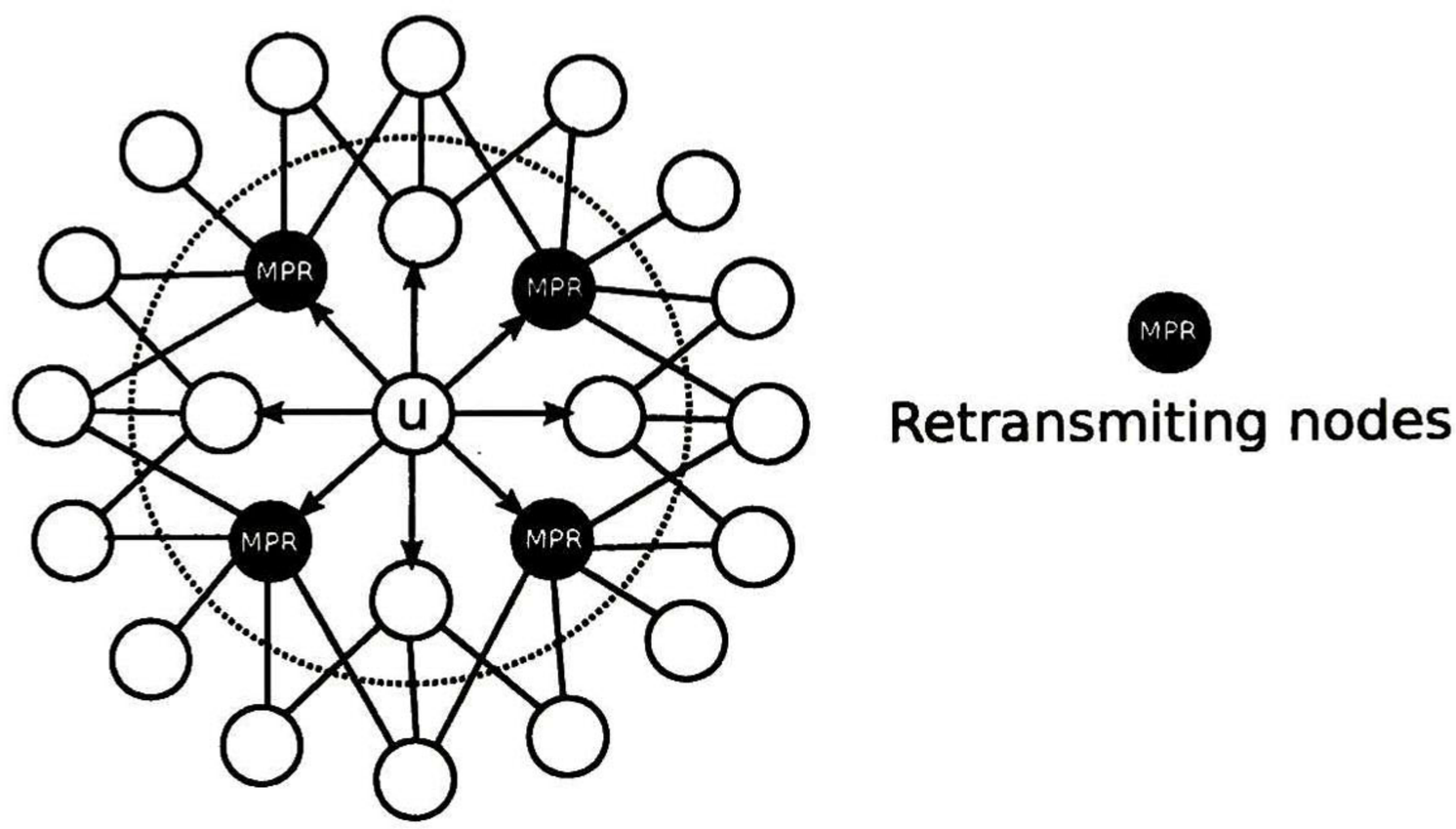

Figure 1.1: MultiPoint Relay - Only the MPR nodes retransmit the packets

1.3 Connected Dominating Set

Connected Dominated Set (CDS) is used in a variety of different applications, especially at the lower levels of the network protocol stack. Some of the applications deal directly with the topology of wireless ad-hoc networks (33). Given an undirected graph $G = (V, E)$, a subset $C \subseteq V$ is a CDS of G if, for each node $u \in V$, u is either in C or there exists a node $v \in C$ such that $uv \in E$ and the subgraph induced by C, i.e., $G(C)$, is connected. The nodes in the CDS are called dominators and the others are called dominatees (Figure 1.2).

Dominating sets play an important role in energy saving for individual sensors in wireless sensor networks. Obviously, the smaller the CDS size is, the fewer the retransmissions are. Thus, in order to effectively optimize flooding, the CDS size must be reduced. However, a solution using DS have one major drawback in terms of balanced

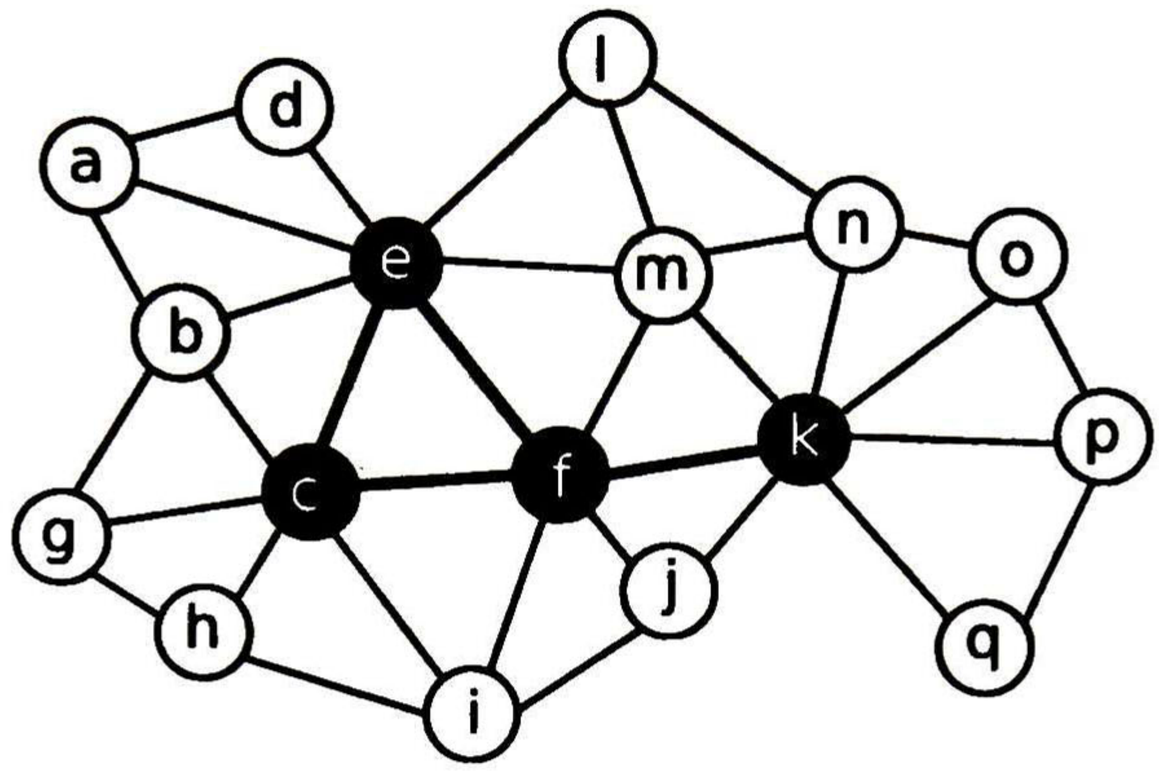

Figure 1.2: Connected Dominating Set

energy distribution among the sensors. This is because the sensors in the dominating set will quickly consume their energy by sensing and transmitting data while the other inactive sensors can save their energy. This will result in disproportionate energy consumption among the sensors, which is not expected in many applications where network lifetime depends on the functioning of individual sensors. In addition, computing a minimum size CDS is also *NP-hard* (18).

The earlier problem is examined by Kamrul et al. (25). The basic idea is to find as many disjoint dominating sets as possible, and use each one for a certain amount of time and then replace it with another one, and so on. If initially all the nodes start with the same amount of energy, then one can expect that the energy consumption will likely be balanced among the nodes. The maximum number of disjoint dominating sets is bounded by the minimum degree of the graph plus one; since a node can either be in one of the dominating sets or be dominated by at least one of its neighbors. In general, it is expected that the size of a dominating set be as small as possible. Even though, the energy distribution is resolved, the environment they use is short of mobility. In this way, if the nodes move around the environment, the strategy does not work. Then, it will be necessary execute the algorithm repeatedly in order to maintain the graph connected, and as a result it will increase its complexity. Aside from mobility problem the nodes have homogeneous transmission power.

Probably the easiest way to create a maximal independent set in any graph is a greedy strategy. Nieberg (33) shows that it is easy to modify a centrally executed greedy

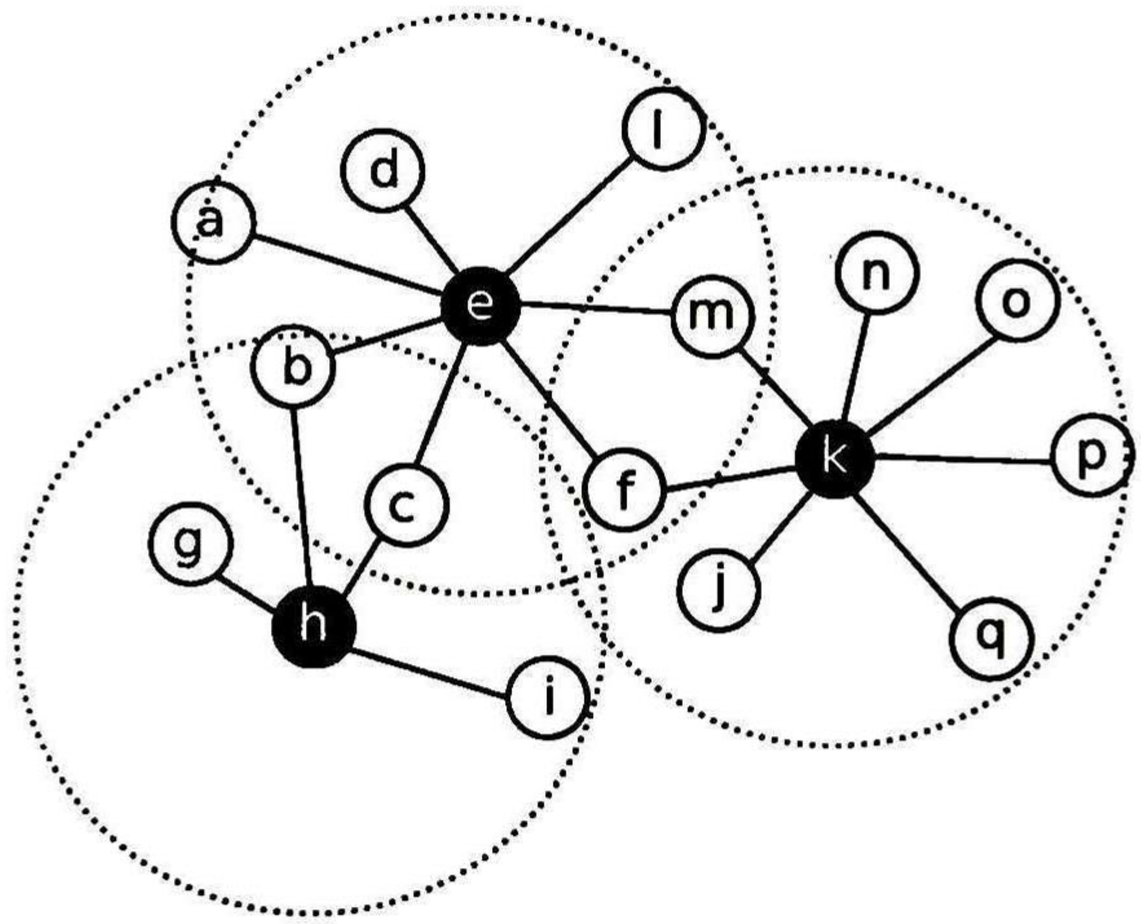

Figure 1.3: Clustering technique

algorithm to run locally, and where each node only needs to interact with its direct neighbors. The local version of the algorithm works as follows: all nodes start out in an undecided state, then an undecided node with the largest weight in its neighborhood, which it does not have a master node, declares itself as master-node. Finally, nodes that learn about a neighboring master node declare themselves as slave-nodes.

1.4 Clustering Approaches

Clustering techniques group nodes into clusters; a clusterhead ensures connectivity between the nodes in the cluster and other clusters (Figure 1.3). There are many proposed clustering schemes for identifying the control nodes that form a virtual backbone among the nodes. In (34) a lightweight clustering protocol for self-organization is presented. It creates a multi-hop, collision-free, and adaptive communication infrastructure. The nodes wake up randomly, and then the nodes that wake up earliest are elected to be cluster-head. Once the nodes are organized as an adaptive multi-hop system, they perform tasks for saving energy. However, a large quantity of nodes is required for creating an organization. Nodes mobility, role change, and fault tolerance are not dealt in this strategy.

The work described above uses diverse techniques, rules, and metrics. A common

feature is that the transmission ranges are not variable. Using variable transmission ranges implies the creation of new rules in the system organization for obtaining energy saving (11). Thus, the nodes can transmit packets to every destination using just the pertinent energy. A good transmission power control algorithm for wireless networks should provide an energy efficient mechanism because devices in these networks are powered by small batteries, and they may be difficult or impossible to replenish frequently.

Wendi et al. (21) presented a micro-sensor networks where the base station is fixed and located far away from the sensors. All nodes in the network are homogeneous and have energy constrains. The authors examine two protocols: direct communication with the base station, and minimum energy multi-hop routing using their sensor network and radio models. By analyzing advantages and disadvantages of conventional routing protocols using their model of sensor networks, they have developed Low-Energy Adaptive Clustering Hierarchy (LEACH); it is a cluster-based routing protocol that minimizes global energy usage by distributing the load to all the nodes at different points in time. LEACH reduces communication energy by as much as eight times compared with direct transmission and minimum transmission energy routing. However, it is necessary to know a priori the number of cluster-heads in the network. In addition, all nodes in the network are homogeneous.

Younis and Fahmy (49)(48) present a stand-alone distributed clustering approach that considers a hybrid of energy and communication cost. Based on this approach, they present a protocol called Hybrid Energy-Efficient Distributed clustering (HEED). It has four primary goals: (i) increase network lifetime by distributing energy consumption, (ii) termination of clustering process within a constant number of iterations/steps, (iii) minimizing control overhead, and (iv) the production of well-distributed cluster heads and compact clusters. This allows identifying a set of cluster heads which cover the entire field. Then, the nodes can directly communicate with its cluster head via a single hop.

Yunseok et al. (28) proposed a transmission power control algorithm for wireless sensor networks called the On-Demand Transmission Power Control scheme (ODTPC). A link quality between a pair of nodes is measured after the sender and the receiver exchange data-ACK packets rather than measuring link quality to every neighbor in the initialization phase. There is no additional packet exchange to maintain good link

quality, and adjust the transmission power level. However, they are not creating a topology.

Most of the previous works suppose the entire network is known. Therefore, each strategy requires a lot of resources. For example, finding the minimum CDS in unit disk graph has been shown to be NP-hard (18), as well as finding the optimum MPR. This is the main reason why we propose the use of self-organization strategies to alleviate the clustering problem.

1.4.1 A Self-organization Approach

In the design of wireless networks, it is often necessary to connect the whole network without using a lot of resources. In order to interconnect a set of nodes, we propose the use of a self-organization strategy. This strategy allows us to reduce complexity topology construction, and energy consumption. Perhaps the most obvious and most interesting characteristic of the self-organization strategy is a property called *emergence*. Complex patterns on the system were not written into the simple rules of the algorithm that produce them. The rules determine the role a node will take depending on the role of its neighbors; but, these never define the structure the network will have. It emerges somehow from a lower level specification of the system.

1.4.2 Construction of the Network

Most of the approaches in the literature are global in nature, i.e., each node must maintain knowledge of the complete network. Indeed, most of the previous strategies do not support adaptive reconfiguration, i.e., the clustering level cannot be changed until the new configuration is made by the network leader. Therefore, the existing algorithms are not adaptable to various node distributions or to different sensing areas. If the sensing area is changed by dynamic circumstances of the networks, the fixed-cluster algorithms may operate inefficiently in terms of energy consumption.

Flooding the whole network with broadcast messages creates the so-called broadcast storm problem (43). This results on excessive collisions, i.e., a large protocol overhead. Using clustering to form a virtual backbone to propagate flooding messages allows us to overcome this problem by reducing the number of messages. Furthermore, the nodes in the virtual backbone can be used to efficiently reach the entire network by broadcasts from only these nodes.

Cluster-based control structures allow a more efficient use of resources. A hierarchical view of the created network through clustering decreases the complexity of the underlying network. Especially in sensors networks which are expected to consist of large amounts of individual nodes. In this way, clustered structures can make a highly mobile topology appears more static, and thus mitigate the effects of mobility.

On a topology level, clustering is usually done by grouping nodes inside certain transmission area, which are then controlled by a designed leader node. This leader node is selected according to a weight, which may correspond to a node's capability to perform additional duties. It can be determined by taking into consideration aspects like residual energy of a node, its memory amount, processing capabilities, the number of neighbors, etc. Usually, these weights are computed locally in each node, and they may depend on the application that the structure is used for. In this case, the weight is a function of the residual energy and the number of neighbors.

Clustering offers several advantages in mobile ad hoc networks. First, network clustering improves routing and mobility management. It increases system capacity, reduces signaling and control overhead, and minimizes network congestion. This makes the network more scalable, and enables support for larger network sizes. Second, clustering stabilizes the network topology and provides a virtual infrastructure for a dynamic network. Here, the clusterhead acts as a base station for its cluster. Third, clustering helps to perform a better resource allocation (13).

In order to get more benefits from clustering, it is important to keep clusters as stable as possible. Frequent changes in the clustering architecture can cause high communication overhead. Stable clustering architecture reduces the frequency of clusterhead changes. That is why we are proposing a self-organized distributed algorithm which is based on groups. This approach yields properties such as *scalability*, *robustness against segmentation* and a *complex behavior*.

In this thesis, a node is modeled as an agent. For us the terms node and agent will be exchangeable. Now, under this model, it is assumed that:

- Every agent has a unique identifier ID e.g. its IP address.

  Every agent knows their neighbors to one-hop.

- The agents can move, arrive, or leave the network.

Each node can adjust its transmission power according to the neighbor's role for energy conservation.

The agent can use overhearing to obtain important information for reducing the message transmission.

- Every agent keeps: a neighbors table and a weight that is equal to the product between the number of neighbors N and the residual energy units e ($weight = N_u * e_u$).

  The agents do not know their geographical positions.

- The wireless nodes are placed in the 2-dimensional Euclidean plane (it also works correctly if agents are located in three-dimensional space).

For addressing the above described problem the proposed algorithm has the following properties:

- It is an approximated decentralized algorithm.

  It operates in a localized way.

- It has low processing complexity.

  It allows network reconfiguration coping with node's mobility.

Elements of each group play different roles such as *leader*, *gateway*, *bridge* and *member*. Each group is composed by one agent playing the role of leader, zero or more agents playing the role of gateway and bridge, and one or more agents playing the role of member. The leader makes possible the communication among members of its group or different groups. A gateway device is responsible of communicating members of different groups through leaders of groups; bridge nodes connect different segments of the network. Finally, members are connected to a single leader (see Figure 1.4).

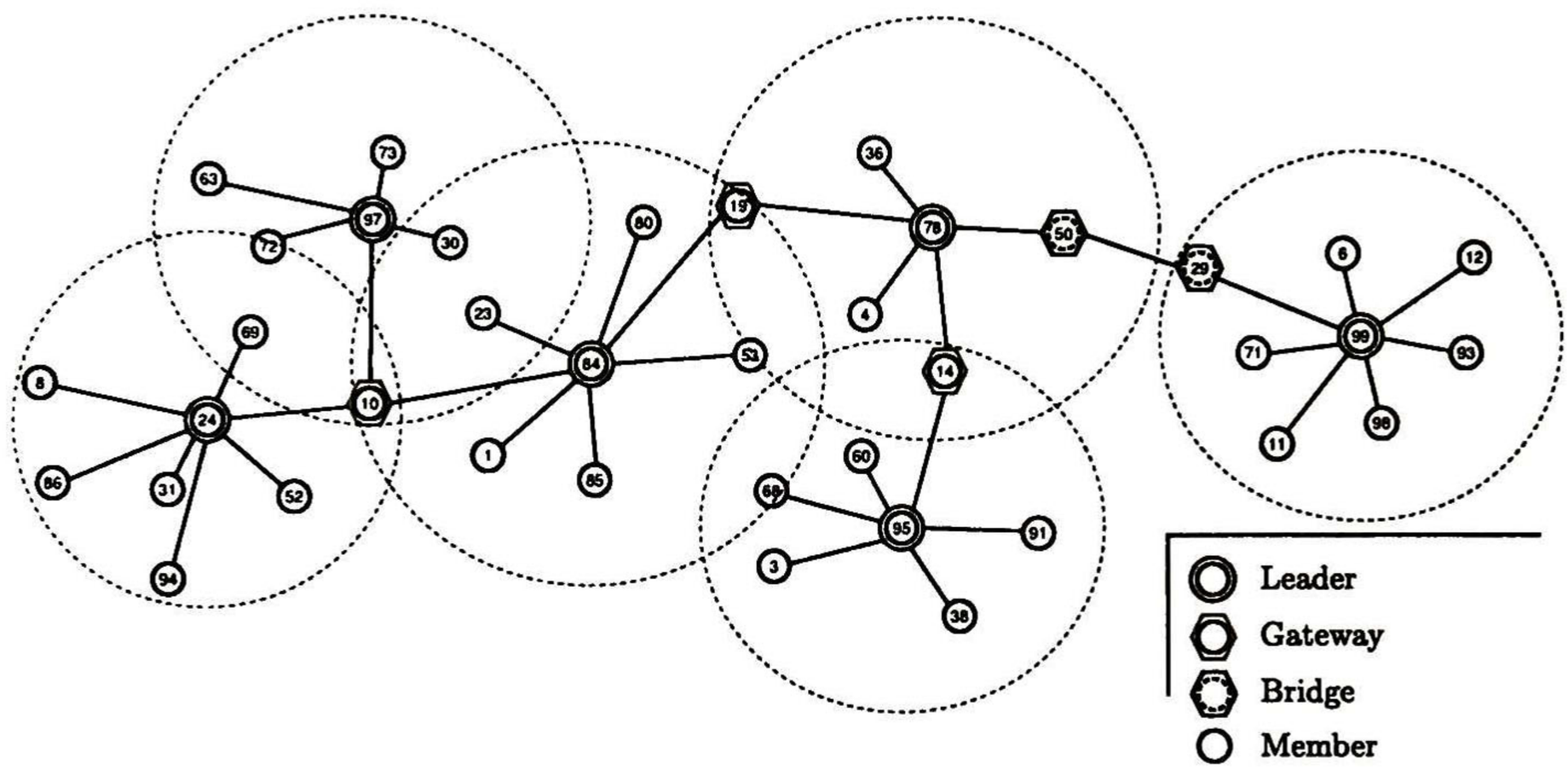

Figure 1.4: Structure of the network

Chapter 2

Self-Organized Algorithms

Abstract. This chapter presents a first approach for coping with problems previously mentioned about wireless ad-hoc networks. First, self-organization strategies are proposed and how they are applied in these networks is described. Description of the algorithms and network modeling is described.

2.1 Problem Statement

Some general terms and definitions that will be used frequently throughout this work are described.

2.1.1 Graph Model for Wireless Topology

The problem can be studied using a graph representation of the wireless ad-hoc network, where edges correspond to the connection between nodes. There are a large number of optimization problems in graph theory. They are important when dealing with efficient wireless communications strategies. It is known that a broadcasting based on minimum spanning tree consumes energy within a constant factor of the optimum.

We can study the network structure as Disk Graphs (33) where all the edges in the network are bidirectional, called Disk Graph with Bidirectional Links (DGB).

We assume that nodes are deployed on the plane and a wireless ad hoc network is modeled by an undirected graph $G = (V, E)$, where V is the set of agents, each

equipped with a radio for wireless communication. $E \subset V \times V$ is the set of possible interconnections between pairs of agents. For each edge $(u, v) \in E$, there is a weight $w(u, v)$ that represents the cost to connect the agent u and v.

We assume that we have a connected, undirected graph G with a weight function $w : E \rightarrow \Re$. Then, by using Prim's algorithm (8), we find a Minimum Spanning Tree T for G ($T \subset G$). We want to use the MST as a comparison of the proposed strategy.

We first provide a rule to calculate the *weight* function. Here. N_u is the number of neighbors of the agent u, and e_u is the residual energy in a time. If $(u, v) \notin E$, then $w(u, v) = \infty$, $c(u, v) = N_u * e_u$ is the cost of link (u, v) (note that $c(u, v)$ is different to $c(v, u)$). Thus, the *weight* function is calculated in the following way:

$$w(u, v) = \frac{1}{c(u, v) + c(v, u)} \tag{2.1}$$

Definition 2.1.1. Let T be a subgraph of a graph G, such that it has been generated by MST (Prim's algorithm). A *backbone* of T denoted by MST_B^T is defined as the set $MST_B^T = T \backslash S$, where $S = \{v \in T | deg_T v = 1\}$. and $deg_T v$ denotes the degree of vertex v

Having heterogeneous transmission ranges causes the graph to be directed. For this reason, we are considering a solution induced only by the symmetric edges. That is, two nodes are connected only if their separation is less than the minimum of their transmission ranges. A edge $(u, v) \in E$ if and only if $d(u, v) \leq min\{r_u, r_v\}$, where $d(u, v)$ denotes the Euclidean distance between u and v, and $r \in [r_{min}, r_{max}]$ is the transmission range.

In next section is shown that our algorithm is close to the minimum spanning tree. Although a MST is a sparse connected subgraph. it is often not considered a good topology since close-by agents in the original graph G might end up far away. Now. if mobility is considered, it is argued that an algorithm should not only be distributed, but even local. If the nodes are frequently moving. it is necessary to consider redundant links in the virtual backbone in order to maintain the network connected. Even more. having redundant links allows reconfiguring the virtual backbone without using a lot of energy.

2.1.2 Definitions

Definition 2.1.2. A communication graph $G = (V, E)$ is composed by a finite set of vertices V representing the wireless nodes and a finite set of edges E representing the nodes communicating directly.

Definition 2.1.3. An Independent Set is a subset of mutually nonadjacent vertices. A *Maximal Independent Set* is an independent set such that adding any vertex not included in the set, breaks the independence property of the set.

Thus, any vertex outside of the maximal independent set must be adjacent to some node in the set.

Definition 2.1.4. The Dominating Set S, is a subset of V such that each node in $V \backslash S$ is adjacent to at least one node in S.

Thus, every MIS is a dominating set. However, since nodes in a dominating set may be adjacent to each other, not every dominating set is a MIS.

Definition 2.1.5. A Connected Dominating Set (CDS) is a dominating set of G which induces a connected subgraph of G (3).

Definition 2.1.6. The neighborhood of a vertex u. $N(u)$ is defined as the set

$$N(u) = \{v \in V | (u, v) \in E\}. \tag{2.2}$$

Then the set of links in the neighborhood of u is

$$\Gamma(u) = \{(u, v) | v \in N(u)\}. \tag{2.3}$$

Using the previous definition, the concept of two-hop group can be defined.

Definition 2.1.7. The two-hop group of a vertex u is defined as the set:

$$C(u) = \cup_{v \in N(u)} N(v). \tag{2.4}$$

Then the set of all links in the two-hop group $C(u)$ is

$$\Gamma(C(u)) = \cup_{v \in C(u)} \Gamma(u). \tag{2.5}$$

Definition 2.1.8. The set of leaders for a node u is defined as the set:

$$N_L(u) = \{v \in V | (u, v) \in E, v \in I\}. \tag{2.6}$$

$\delta(G)$ and $\Delta(G)$ denote the minimum and maximum vertex degrees in G respectively, and $\delta_C(G)$, $\Delta_C(G)$ denote the minimum and maximum two-hop vertex degrees in G respectively.

2.2 Group-based Strategy

We are proposing a group-based distributed algorithm using a self-organization strategy that manages the assignment of four different roles to be played by the agent (35). The procedure derived from the algorithm is executed by every agent in the network. We express a desired global behavior as a set of local rules to be followed by each entity. This system architecture will lead to a scalable and robust system that follows the desired global behavior.

2.3 MWAC-based algorithm

Initially, when the agents wake up, they do not have assigned any role. The first role assignment is leader for all the agents. Then, a conflict emerges, which is solved by a leader-election procedure. Some agents play the role of members, and other agents become gateways. The bridge agent is connected to other bridge agents in order to fix a possible segmentation. In this way, clusters are formed.

In order to avoid segmentation in the network, we proposed a bridge agent. When a member agent detects another member agent that belongs to another group, it will try to make a connection between them. Consequently, they will switch their role to bridge agents.

2.3.1 Notation and Functions

The following list provides the necessary elements (notation and functions) to describe the self-organizing algorithm.

- Events

  $BR(u)$; *bridge request*

  $BA(u)$; *bridge acknowledge*

  $IR(u)$; *check inconsistency*

- Sets

  $N(u) = \{v \in V : (u, v) \in E\}$

$N_L(u) = \{v \in N(u) : Role(v) = Leader\}$

$N_G(u) = \{v \in N(u) : Role(v) = Gateway\}$

- Cardinalities

$deg(u) = |N(u)|$; *it is the number of neighbors*

$deg_L(u) = |N_L(u)|$; *it is the number of leader neighbors*

- Functions

$weight(u) = deg(u) * e_u$; *where* e_u *is the residual energy of* u

$id(u)$: *it returns the agent's identifier*

$id_g(u)$: *it returns the* u*'s group identifier*

$Role(u)$: *it returns the current role of the node* u

- Predicates

$C_0(u) = true$ if $\exists v \in N(u) : Role(v) = Leader$

$C_1(u) = true$ if $deg(u) > 0 \wedge \neg C_0(u)$

$C_2(u) = true$ if $\exists v \in N(u) : Role(v) = Member \wedge id_g(u) \neq id_g(v)$

$C_3(u) = true$ if $\exists v \in N(u) : Role(v) = Gateway \vee Role(v) = Bridge$

$C_4(u) = true$ if $\exists v \in N_G(u) : N_L(u) = N_L(v) \wedge id(v) > id(u)$

$C_5(u) = true$ if $\exists v \in N_G(u) : N_L(u) \subset N_L(v)$; *where* $\subset$ *is the proper set*

if the conditions do not hold then $C_i(u) = false$

2.3.2 Rule Base

The strategy that defines and updates the agent's role is embedded in the following reduced set of rules.

$R_1 : deg(u) = 0 \Rightarrow (Role_u - Any)$ //Any is the default value for role

$R_2 : C_1(u) \Rightarrow (Role_u - Leader)$

$R_3 : deg_L(u) = 1 \Rightarrow (Role_u \leftarrow Member)$

$R_4 : deg_L(u) > 1 \Rightarrow (Role_u \leftarrow Gateway)$

$R_5 : C_2(u) \wedge \neg C_3(u) \Rightarrow BR(v)$

$R_6 : BA(v) \wedge \neg C_3(u) \Rightarrow (Role_u \leftarrow Bridge)$

$R_7 : (C_0(u) \wedge Role(u) = Leader) \Rightarrow IR(v)$

$R_8 : IR(v) \wedge weight(v) > weight(u) \Rightarrow (Role_u \leftarrow Member)$

$R_9 : Role(u) = Member \wedge BR(v) \Rightarrow (BA(v) \wedge Role_u \leftarrow Bridge)$

$R_{10} : Role(u) = Gateway \wedge C_4(u) \Rightarrow agent_u\ turn\ off$

$R_{11} : Role(u) = Gateway \wedge C_5(u) \Rightarrow agent_u\ turn\ off$

2.3.3 The Procedure for Obtaining the Virtual Backbone

At start-up, each agent broadcasts *hello* messages in order to discover their neighborhood. When an agent realizes the neighbor's table has not changed for a period of time σ, it will increase the broadcast transmission interval T by a constant Δ in order to save energy.

When an agent has no neighbors, rule R_1 is applied; then the agent acquires the role *any*. After an agent detects some neighbors; there is more than one rule to follow, depending on the neighbors' roles. When in the agent's neighborhood there is no leader, rule R_2 is applied and this agent acquires the role of *leader*. An agent becomes a member when either R_3 or R_8 is applied, i.e., when either it detects only one leader in its neighborhood or it loses its leadership due to its low *weight*. Whenever an agent detects more than one leader, it changes its role from *member* to *gateway*, through the application of rule R_4.

2.3.4 Leader Role

The most important role of an agent in the network is leader. Leader agents are responsible for carrying out all the communication in the group. For this reason leader agents must have the highest weight, i.e., they cover most of the agents in the neighborhood and have the highest residual energy. For computing the agent's weight, we consider

only the number of neighbors and the residual energy ($weight(u) = deg(u) * e_u$). This weight, however, could change according to the environment we are considering.

Initially, when an agent discovers at least one neighbor, this agent will seek a leader among its neighbors; if it does not find one, it becomes the leader (R_2). After that, when agents take on some of the available roles, a conflict could arise because two or more agents are leaders in the same group; when this situation happens (R_7), it is necessary to reach an agreement to select only one leader. In order to keep the role of leader, the agent must have the highest weight in the list of the conflicted agents (R_8).

2.3.5 Bridge Role

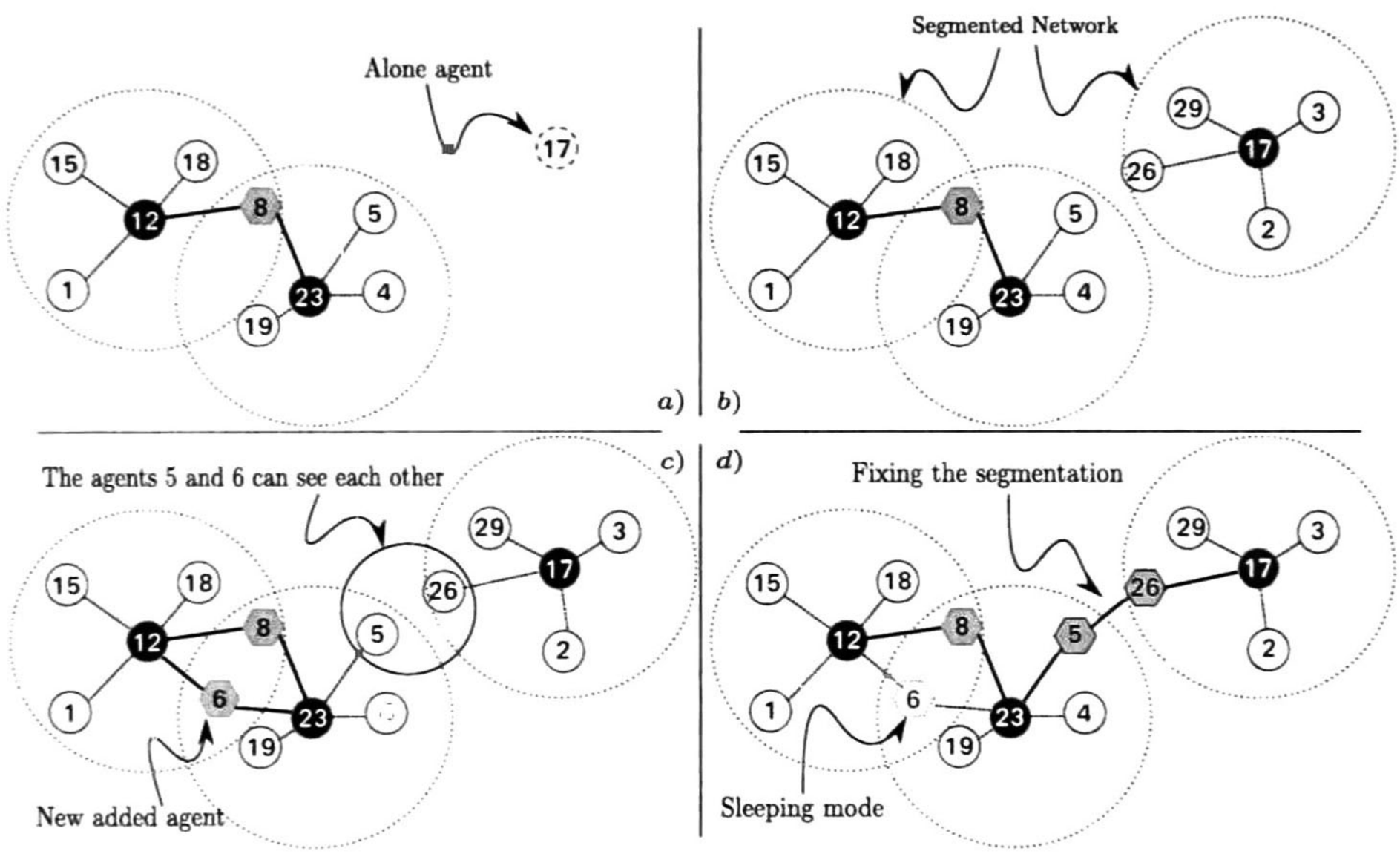

Figure 2.1: Construction process - Clustering process of network

In order to carry out the *bridge* role, the agent must satisfy certain requirements. It is difficult to know which agent must be a *bridge* agent, because agents are aware only of their own neighborhood, i.e., the agents do not perceive other agents beyond their one-hop neighbors. One of the objectives of this algorithm is to maintain complete connectivity; however, the network may be created in a segmented way (see Figure 2.1b).

A solution to solve the segmentation problem could be to start a process of reorganizing any of the segments in the network, i.e., every *member* that detects an inconsistency must inform its leader about the situation. Then the *leader* agent searches for the group's *id* involved in the segmentation problem. After that, if the *leader* agent does not find this *id*, it will start the reorganization in order to alleviate the segmentation. However, every *leader* agent that detects the inconsistency will execute this procedure; this is both time-consuming and energy-consuming. The previous solution not only wastes a great deal of energy, due to excessive message usage; it also fails to ensure good reorganization and overall connectivity in the affected sub-network. To avoid these shortcomings, it is preferable to use *bridge* agents.

Every time a *member* agent satisfies the rule R_5, it generates an event that involves the sending of a bridge request message to v ($BR(v)$). After that, when a *member* agent receives a bridge request from v, it becomes a *bridge* and returns a *bridge* acknowledge message to v (R_9). Now, when a member agent u receives an acknowledge message from v, it becomes a *bridge* agent, as shown in Figure 2.1d.

2.3.6 Gateway Role

The task of the *gateway* agent is to communicate with the *leaders*. In most cases, more than one *gateway* communicates with the same set of *leaders*. In order to alleviate the redundancy problem, a *gateway* agent will turn inactive if either rule R_{10} or R_{11} is satisfied. Let us suppose a *gateway* u. This *gateway* turns inactive if the set of *leaders* in its neighborhood is equal to the set of *leaders* of one *gateway* neighbor v, and at the same time, the *id* of u is lower than the *id* of v, or the set of *leaders* of u is included within the set of *leaders* of one gateway neighbor.

2.3.7 Energy Management

The self-organization approach dynamically adjusts the transmission range of each agent in order to improve network connectivity. Each agent u can send its information to all agents within its transmission range by one-hop broadcasting.

Some *member* agents do not need to use the maximum transmission power to communicate with their *leaders*. It is better for a *member* agent to adjust its transmission power just to reach its *leader*. In this manner, a *member* agent uses just the necessary

energy (Figure 2.2). In order to calculate the optimal transmission power, we are using the following equation (9):

$$R_{TX_{min}} = \frac{RX_{threshold} * P_{TX}}{P_{RX}} \tag{2.7}$$

where

- $R_{TX_{min}}$ is the necessary transmission power to communicate with a leader agent.
- $RX_{threshold}$ is the minimum signal strength to receive the packet. If one packet is received whose signal strength is stronger than $RX_{threshold}$, the packet is received correctly, otherwise it is discarded.
- P_{TX} is the transmitted power.
- P_{RX} is the received power.

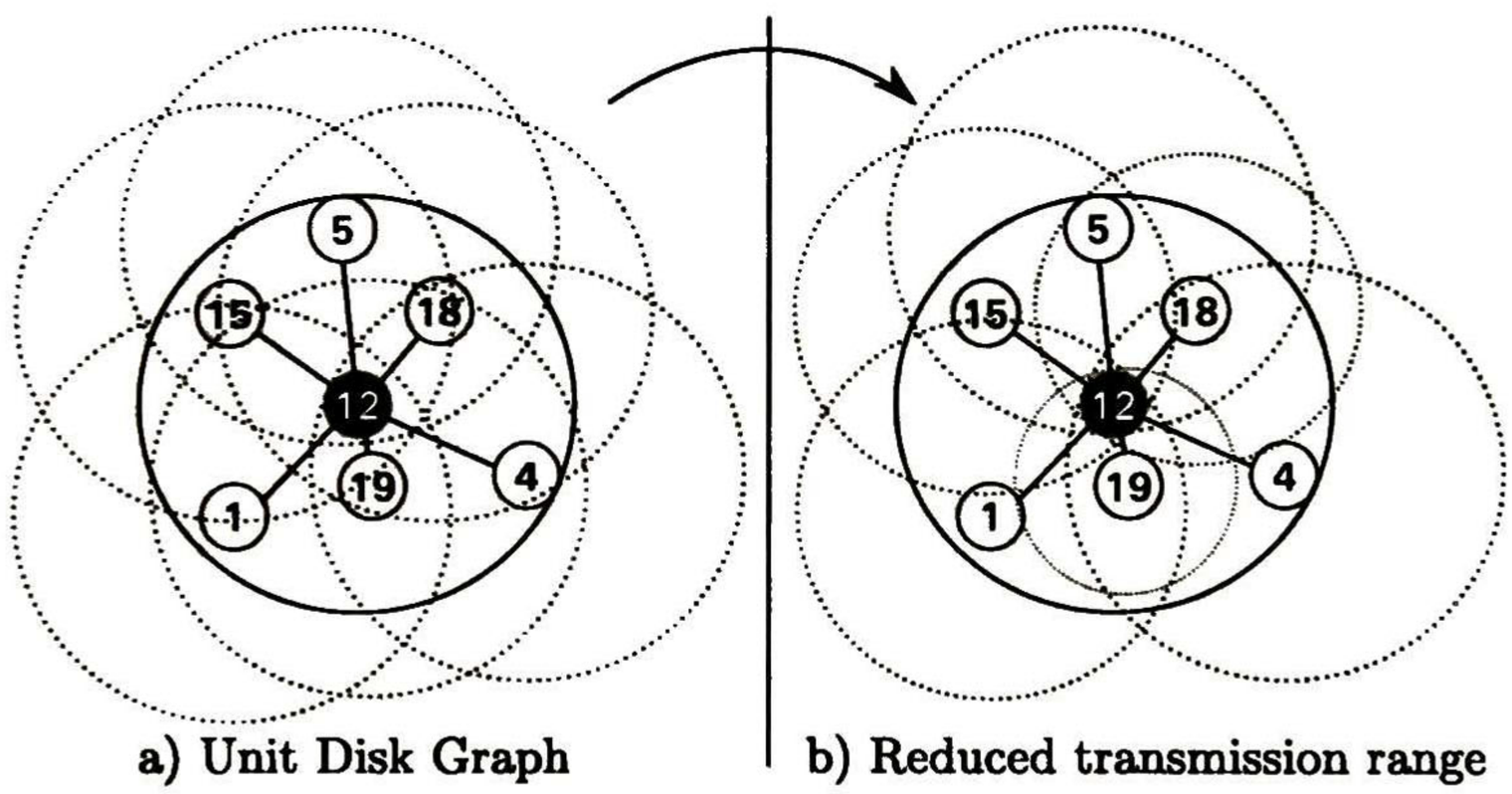

Figure 2.2: Reduction of transmission range - Energy saving by reduction of transmission range

2.4 An Energy-Efficient Self-Organized Algorithm EESOA

2.4.1 Description of the algorithm

The structure generated by the proposed algorithm is based in groups, which it is composed by nodes that may adopt four different roles (leader, gateway, member and

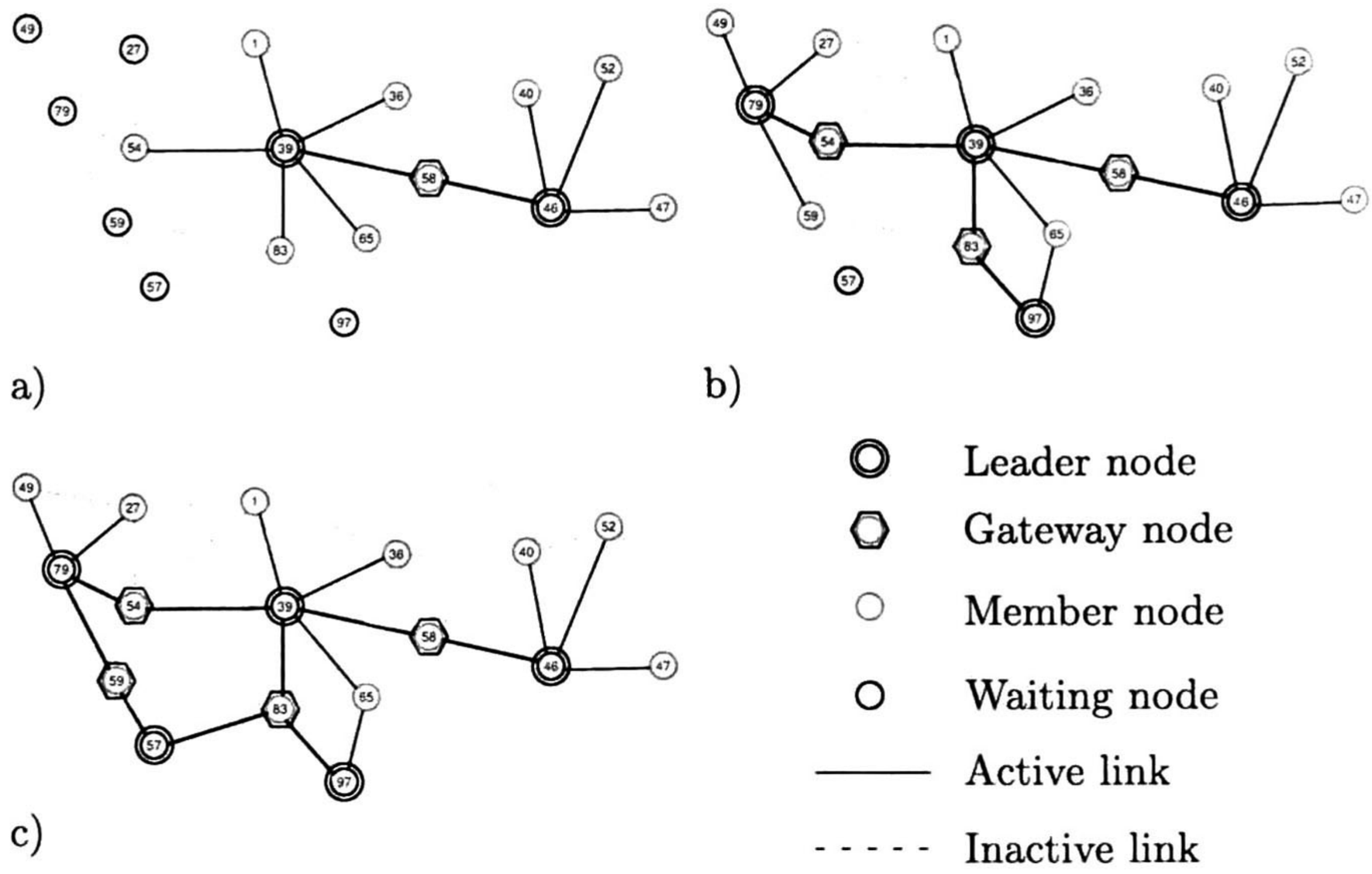

Figure 2.3: Virtual backbone step by step - a) first step, b) second step. c) full connected network.

bridge); there is only one leader per group. Leaders are in charge of doing all the communication inside the group they coordinate. Thus, it is necessary to choose the more suitable nodes to play the leader role. The gateway node makes the communication between the leaders; it can communicate more than two leaders. In this way the virtual backbone is formed with leaders and gateways. Then, the communication is carried out by mean of these nodes. The member node only takes care of its own tasks, and there can be zero o more members connected to one leader. Figure 2.3 shows the backbone formation under this strategy.

Because the algorithm is localized as well as the MWAC-based algorithm, it is impossible to determine a whole connection by using only leaders and gateways as backbone. Figure 2.4d shows a structure where the backbone does not connect the whole network with only leaders and gateways, because of communication between members is not defined; this situation is called segmentation. By using bridge nodes, the different network segments can be connected.

Using the *hello* message, the proposed algorithm is able to build the backbone by

means of inhibiting the neighborhood. This is done in the following way, whenever a *hello* message is sent by a leader, it will inhibit all nodes that receive the message. Whenever a new node arrives to the network, if it receives a *hello* message from a leader, the node will be inhibited and become member. If no *hello* message from any leader is received, the node stars discovering its neighborhood, then it follows the proposed algorithm.

A node that is not inhibited by a neighbor will get the leader role as long as it has the greatest weight in the group, with ties broken by the node *id*. A node will wait if it has an uninhibited neighbor either with a greater weight or with the same weight but with a greater *id*. After a node u takes the role of leader, it broadcast an inhibiting message. After receiving this message, all the $u's$ neighbors will be inhibited. A node will switch to gateway as long as it is inhibited by more than one leader. If a node is inhibited by only one leader, then it takes the member role. This behavior is described in the Algorithm 1.

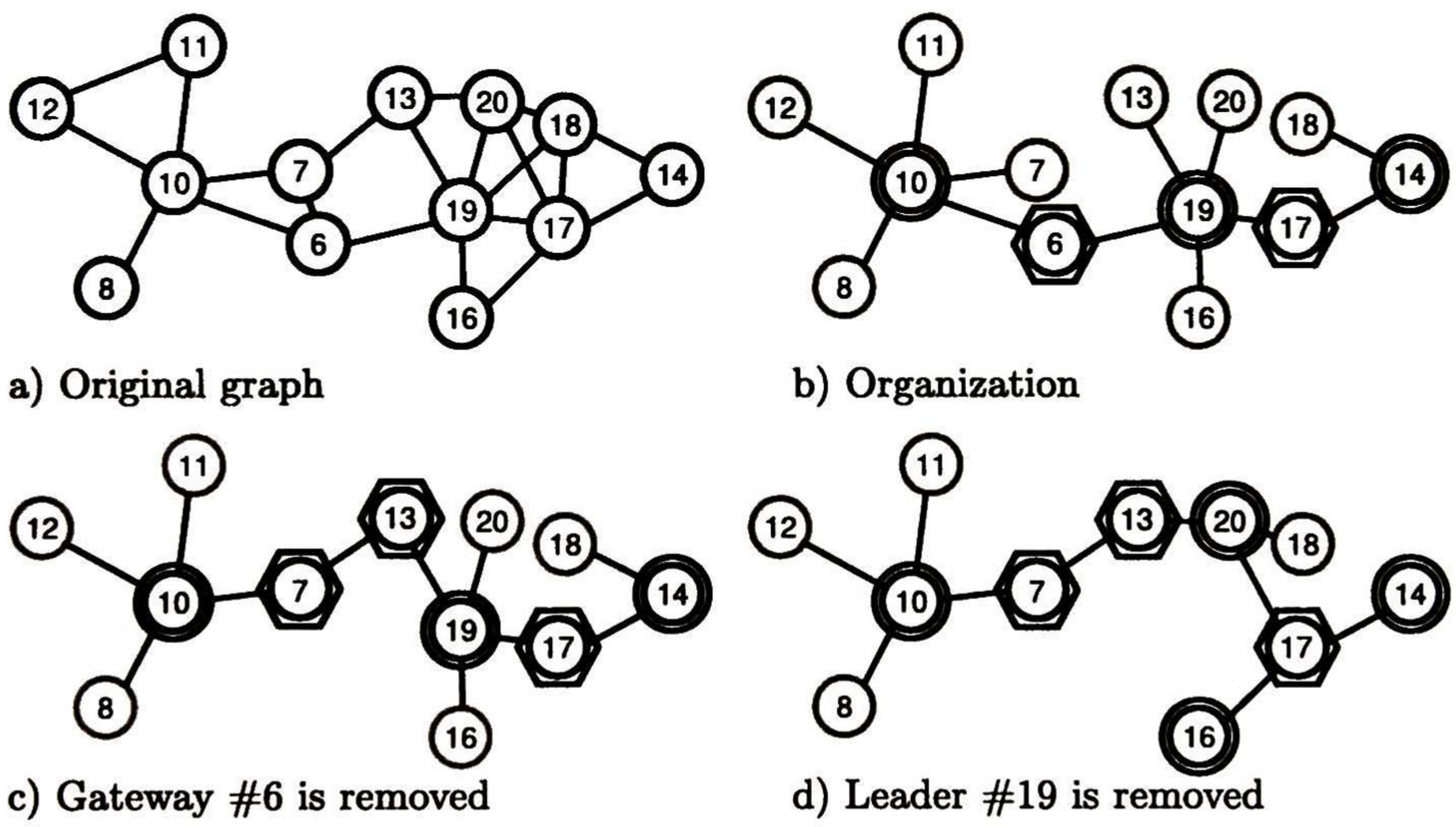

Figure 2.4: Virtual backbone reorganization

The segmentation problem can be solved by using bridge nodes. In order to connect different segments of the network, a member node u will get the bridge role if it fulfills the following conditions:

Algorithm 1 Energy-Efficient Self-Organized Algorithm

```
N(u) = {v ∈ V | (u, v) ∈ E}
Ω(u) = {v ∈ N(u) | inhibit_v = false}
X(u) = {v ∈ N(u) | role_v = leader}
procedure EESOA
    if N(u) > 0 then
        if inhibit_u = false, role_u ≠ leader then
            if ∀v ∈ Ω(u), W_u ≥ W_v then
                if ∄v ∈ Ω(u), W_u = W_v, id_u < id_v then
                    role_u ← leader
                    send_pkts(INHIBITED, N(u))
                end if
            end if
        end if
        if inhibit_u = true then
            if |X(u)| = 1 then
                role_u ← member
                TREAT_CLUSTERS
            else
                role_u ← gateway
                TREAT_GATEWAYS
            end if
        end if
    end if
end procedure
```

- u has a neighbor v with the role of member or bridge and the $v's$ group-id is different of $u's$ group-id.
- There is not a bridge node in the neighborhood of u.
- There are not gateway nodes w connecting a leader node with the id equal to the $v's$ group-id.

Each member node has a leader in its neighborhood, and most of the time these member nodes are at the edge of segments, i.e., these member nodes can see each other even though there is no a defined connection between them. That is why member nodes are those that turn to bridge role. In this way, bridge nodes connect segments where possible. These steps are described in Algorithm 2.

Algorithm 2 Cluster Managing Algorithm

```
1:  ID_u is the group-id of u
2:  function TREAT_CLUSTERS
3:      if ∄v ∈ N(X(u)) | role_v = bridge then
4:          if ∃v ∈ N(u) | ID_v ≠ ID_u, role_v = bridge, conn_v = u then
5:              role_u ← bridge
6:              conn_u ← v
7:          else
8:              for v ∈ N(u) | ID_v ≠ ID_u, role_v = member do
9:                  if ∄w ∈ N(u) | role_w = gateway, ID_v ∈ X(w), v is available then
10:                     role_u ← bridge
11:                 end if
12:             end for
13:         end if
14:     else
15:         u is not available for being bridge
16:     end if
17: end function
```

As it can be seen in Figure 2.4, the result of removing the gateway node 6 from Figure 2.4b is the backbone shown in Figure 2.4c. When the node 6 disappears from the network, then member nodes 7 and 13 realize that the conditions to be a bridge node are satisfied. Afterwards, these nodes switch to bridge role to fix a possible segmentation.

2.4.2 Neighbors Table Management

Neighborhood management essentially has three operations: insertion, erasure, and updating. Each node broadcasts *hello* messages to discover its neighborhood. Since all these nodes wake up at the same time, EESOA attempts to schedule the instants at which nodes send their broadcast messages, so that not all nodes send their messages at once. This scheduling helps to avoid collisions. In order to do this, a random time ι between 0 and ξ is computed when the node starts the algorithm. Afterwards, the node will send *hello* messages at intervals of T.

Each node keeps a frequency counter for each entry in its own table. On insertion, counter entries referring to neighbors in the node's table are set to ρ. As the messages of the local node neighborhood reach the node, new discovered nodes will be inserted in the table as neighbors, if there is bidirectional communication between them. This means that the insertion is done when the node receives an acknowledgment from another node, which received its *hello* message. In every cycle or unit of time, the count of all entries is decremented by one. Each time that the node receives a message from a node in the neighbor table, it will reset the neighbor value count to ρ. As time passes, if the count of some neighbor reach zero, it will be dropped out by erasing its entry in the neighbors table.

At the beginning, each node starts the algorithm when the neighbors table has not received a new node entry for a period of $\sigma * T$ rounds.

2.4.3 Increasing Network lifetime

2.4.3.1 Redundant Gateways

Most of the time more than one gateway connects the same leaders. It is obvious that duplicate gateways will affect the overall energy consumption, and consequently the network lifetime is reduced. Sometimes, in order to establish communication among a set of leaders it is only necessary one gateway. When the gateway node depletes its energy, it is no longer able to communicate its leaders. In this way, another gateway must be available to maintain the communication between leaders.

This situation can be described as follows: Suppose that w is a gateway, if the leader set of w is a subset of other gateway's leader set, which is inside its neighborhood, it will switch to member node. If the $w's$ set of leaders is exactly the same as other gateway's

leader set, and id_w is lower than the other one, then w will switch to member node as well (see Algorithm 3).

Algorithm 3 Redundant Gateways Algorithm

```
function TREAT_GATEWAYS
    for v ∈ N(u)|role_u = gateway do
        if X(u) ⊆ X(v) then
            if X(u) = X(v) then
                if id_u < id_v then
                    role_u ← member
                end if
            else
                role_u ← member
            end if
        end if
    end for
end function
```

Gateway nodes that left their role, i.e., gateways that turned to member due to redundancy, will join to the leader with the strongest signal strength, $max\{P_{RX_v}|v \in Nl(w)\}$. Finally, gateway nodes that switched to member cannot take the role of bridge.

2.4.3.2 Reorganization

Often changes can happen in the environment due to new obstacles, movement of nodes, starvation or failure of a node, increasing the loss rate of a link, etc. Therefore, when a node dies, reorganization must be carried out. A complex situation appears when leaders die or move because they are in charge of the communication in the group. Figure 2.4b shows a piece of network including thirteen nodes. Let us suppose the gateway with id number 6 dies; when node 7 detects the gateway is missing, it makes the communication through node 13 by using the bridge role. Figure 2.4d shows how the reorganization is carried out if the departed node is a leader.

2.5 Common properties for MWAC-based and EESOA

2.5.1 Sensor Networks

Sensor networks in event detection have long periods of inactivity; such networks are also expected to be dense, and thus their redundancy can be exploited, allowing several agents to turn off their radio to conserve energy. In these networks, connectivity needs to be maintained in such a way that if an agent with inactive radio senses something of importance; it can become active, and then successfully transmit the information. In this case, if member agents have to sense and send the information after long periods of time, it is not necessary to keep them awake. It is better to turn the member agent to sleep mode to save energy and increase the network's lifetime.

A *gateway* agent maintains the communication among *leader* agents, i.e., it maintains the communication among groups in the network. Usually, in order to maintain maximum connectivity in the network with the lowest power consumption, the redundant *gateway* agents should go into inactive mode. If one active *gateway* agent runs out of energy, one of the inactive *gateway* agents must be awakened instantly to replace the dying agent. This agent scheduling algorithm guarantees overall network connectivity as long as possible.

As we know, a *leader* agent needs more energy than other agents. That is why it is necessary to design a strategy to maintain its leadership as long as possible, because after a *leader* agent dies, it is necessary to reconfigure its neighborhood. In order to delay this situation, when a *leader* agent has less energy than a threshold α, it will reduce its transmission range down to 50% while it maintains at least two *gateway* or *bridge* agents to maintain communication. Whenever the energy residual of the *leader* agent is less than a threshold β, it will not maintain its role and will have to become a *member* agent.

2.5.2 Broadcast

Since communications dominates the energy consumption, it is necessary to avoid sending messages as much as possible. Therefore, the time interval T between two broadcasts of *hello* messages will depend on the environment. After the neighbors table does not insert a new node in a time elapse $\sigma * T$, the broadcast transmission period will be increased by a constant value Δ once. Therefore, the node will transmit a *hello*

message in the interval $T + \Delta$. Thus, the energy consumption can be minimized. It is worth mentioning that this does not affect the finding of new neighbors since a node that joins the network asks for neighbors.

Different events can change the neighbors table such as: joining of a new node, a node leave the network, failure of nodes or simply the node dies because its energy is over; even a more likely situation is the movement of nodes inside the network. Therefore, if the neighbor table changes the parameters, the broadcast transmission must be restarted.

When the node changes the role, it restarts the transmission power and broadcast transmission interval, and resets the initial values.

2.5.3 Overhearing

In these kinds of networks, by overhearing, a node can recognize the source and the destination of packets transmitted by the neighbor nodes. Based on the overheard information, the algorithm forms the clusters without additional packet transmission overhead such as advertisement, announcement, joining, and scheduling messages. Specifically, we have focused on the minimum power topology problem, where the aim is to assign transmission powers according to the node's role in such a way that all the nodes are connected by bidirectional links, and the total power consumption over the networks is minimized.

In short, the differences between the base algorithm (26) and our strategy are given below:

1. Member nodes can adjust their transmission range.

2. The leader holds the leadership for a longer time by using the strategy of varying its power transmission.

3. Redundant gateways are turned off in order to save energy.

4. All the agents vary their broadcast transmission interval.

5. The bridge role is designed to alleviate the segmentation problem.

2.5.4 Topology Control

In order to generate a network with the desired properties, topology control is used. This technique tunes the range assignment to optimize the energy consumption without changing the network structure.

For example, a node w having a gateway role will adjust its transmission power by computing the necessary power level, then it will choose the maximum value among the leaders ($max\{R_{TX_v}|v \in Nl(w)\}$). In the same way, members can compute the optimal power transmission by using the reception threshold of its leader.

In order to change the transmission power, the nodes must wait a certain time, because if the mobility of nodes is too high, a bad structure can be formed. In this way, whenever a gateway or member node keeps the role for an elapsed time κ, it will execute the procedure for adjusting the retransmission power. In this way, when a node decides to reduce the transmission power the network structure is not affected.

2.6 Strategy for selecting the weight

Most of the time the wireless network is modeled as deterministic links. This may be reasonable and convenient in wired networks. However, in mobile ad-hoc networks, due to the interference, physical distance and the like, links are not completely available. Under the deterministic model, any pair of nodes in a network is either fully connected or completely disconnected. However, in most real applications the deterministic model cannot fully characterize the behavior of wireless links. There is a transitional region where a pair of nodes is probabilistically connected. Such pairs of nodes are not fully connected but reachable via the so called lossy links. As reported in (4), there are often much more lossy links than fully connected links. Therefore, their impact can hardly be neglected (20).

2.6.1 Problem Formulation

For any node i and a node j, e_{ij} represents the link between them. The value p_{ij}, which indicates the packet loss probability in e_{ij} link, is called the reliability of the link between the node i and the node j.

The probability that node i can successfully directly deliver a packet to node j is $(1 - p_{ij})$. This means that, on average, to transmit a packet on link e_{ij}, the packet has

to be retransmitted $1/(1 - p_{ij})$ times. Therefore, the energy consumption and delay of the link will be proportional to $1/(1 - p_{ij})$.

2.6.2 Link Probability Estimation

The wide variation in delivery rates for networks suggests that routing protocols may often choose links that are high enough quality to pass routing packets, but which still have substantial loss rates.

The Window Mean with Exponentially Weighted Moving Average (WMEWMA) estimator is used to compute the link reliability (45).

2.6.2.1 WMEWMA

Input to the estimator include packet arrivals M, and periodic timer events T. Each packet contains a transmitter ID and a link sequence number. Since a lost packet does not generate any message arrival events, every M is equivalent to zero or more packet loss events followed by a packet success event. Let P' be the current estimation, t be the time stamp of the last M event, m be the number of known missed packets based on sequence number difference, and k be a guess on the number of missed packets based on R over a time window between the current T event and the last M or T event.

The tuning parameters are t and α. Let t be the time window represented in number of message opportunities between two T events, and $0 < \alpha < 1$. The algorithm works as follows: P' is only updated at each T event. In the time t between two T events, let r be the number of received messages (i.e. number of $1's$ in M events), and f be the total number of missed packets, l, from all M events and the current T event. The mean $\mu = r/(r + f)$, and $P' = P' * \alpha + (1 - \alpha) * \mu$.

Figure 2.5 shows how the estimator behaves when the probability of success changes as time passes for a certain link. Initially the probability of success is 80%, then after a while the probability changes to 20% and finally it changes to 99%.

In considering the expected number of transmissions of a link, it is important to determine link quality for both directions. Link pairs that are very good in one direction tend to be good in both directions, and pairs that are very bad in one direction tend to be bad in both directions (10). This does not mean that nodes will compute the same weight for the same link.

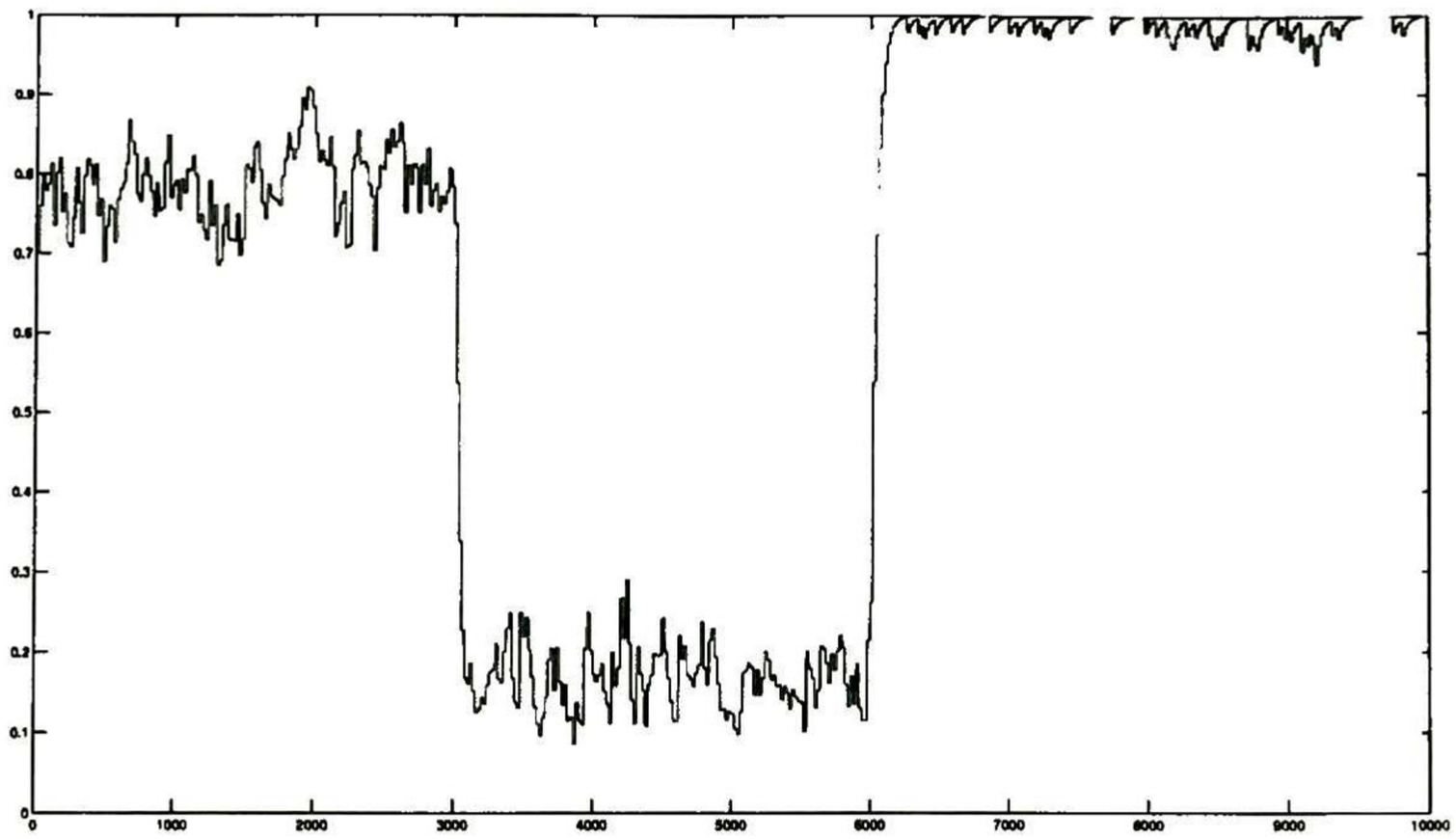

Figure 2.5: wmewma link estimator - Estimation of a link of 80%, 20% and 99% of delivery success.

In order to assign the weight to the link, the nodes will choose the highest value as follows:

$$p = max(p_{ij}, p_{ji})$$

Then

$$w_{ij} = w_{ji} = 1/(1-p)$$

In this way, this estimation can be used to determine the leader weight (w_i) as follows:

$$w_i = \sum_j w_{ij}$$

Afterwards, the group can be formed by choosing nodes that have the lowest w_i value in the neighborhood, i.e., The leaders will be the nodes that have the best connection in the neighborhood.

As described before, there is a way to obtain the weight to know how good a node is based on connectivity. Thus, different metrics can be defined to get a good virtual structure depending on the environment the structure will be used.

Chapter 3

Global Approach

Abstract. This chapter presents two global solutions for setting up an optimum virtual backbone while satisfying connectivity requirements. A multi-objective optimization approach by using particle swarm optimization is addressed for obtaining this optimum virtual backbone.

3.1 Particle Swarm Optimization Approach

Among the existing approaches to solve the problem of networking, one of the most promising is the one based on the property of independent sets (3). In this approach, sensors in a network can be studied as independent sets. Therefore, the goal is to set up an optimum virtual backbone while satisfying the connectivity requirements. Nevertheless, it has been proven that the optimal node selection problem is a NP-complete problem (2).

The independent dominating set problem has been widely studied in the wireless networking community given its applications to ad-hoc routing. However, finding a Maximal Independent Set (MIS) is clearly a combinatorial problem. Therefore, it is necessary to use an approximation method to solve efficiently this kind of problems. One of the most promising techniques for approximation is the Particle Swarm Optimization (PSO). Even though the original PSO was designed for continuous optimization, it can be extended to operate on combinatorial binary search spaces. Furthermore, PSO can

be adapted to work with discrete variables by simply defining the operations to obtain the velocity and the position values of the particles.

3.1.1 Combinatorial Problems

Several works on binary combinatorial problems have been already proposed. For example, Omar et al. (24) introduces a new particle swarm optimizer based on discrete searches for the permutation and combinatorial optimization problem of graphs formation. It is used for describing multi-vehicles systems and distributed mobile networks. The approach addressed in that work considers two planar graphs G_1 and G_2, where the objective is to find a permutation set of G_1 such that the two sets of vertices are brought into correspondence, where the correspondence is defined between vertex sets of G_1 and G_2. Specifically, the problem is the identification of an isomorphism that is topologically equivalent to the desired formation graph, such that the position of each vertex satisfies some optimality condition.

The Steiner Tree Problem (STP) in graphs is one of the most important multiple destination routing problems, which is NP-hard as well. Given a network graph, the goal of STP is to find out a minimal spanning tree that connects the source and all the destination nodes. One of the works in this field was introduced by Wen-Lian et al. (50), in which a novel discrete PSO algorithm was proposed to solve the STP.

Issues of node deployment (46), localization (5) (6), energy-aware clustering (47) (27) (42), and data-aggregation have been formulated as optimization problems as well. An overview of PSO, issues in Wireless Sensor Networks (WSN) and a brief survey of recent PSO-based solutions to the WSN issues are presented in (29).

3.1.2 Proposed Approach

In this chapter, a novel algorithm is introduced to efficiently generate an energy-efficient communication strategy for low-resource large-scale wireless ad-hoc networks. The network is modeled as a graph, and then a PSO is applied for finding the optimal MIS, which provides a global solution for a given ad-hoc network. By using this strategy, a method for calculating an optimal virtual backbone in wireless ad-hoc networks is proposed. Furthermore, it is showed through empirical analysis that the algorithm will produce solutions within a constant factor of optimality.

3.1.3 Ad-Hoc Networking

3.1.3.1 Clustering

In the design of wireless networks, it is often required to connect the whole network using the less quantity of resources. Cluster-based control structures allow a more efficient use of resources because a hierarchical view of the created network through clustering decreases the complexity of procedure of create network. This is especially true in sensors networks composed by a large number of individual nodes.

On a topology level, clustering is usually done by grouping nodes inside a certain transmission area, which are then controlled by a designated leader node (36). Elements of each group play different roles such as leader, gateway, bridge, and member as mentioned in previous chapter (See Figure 3.1).

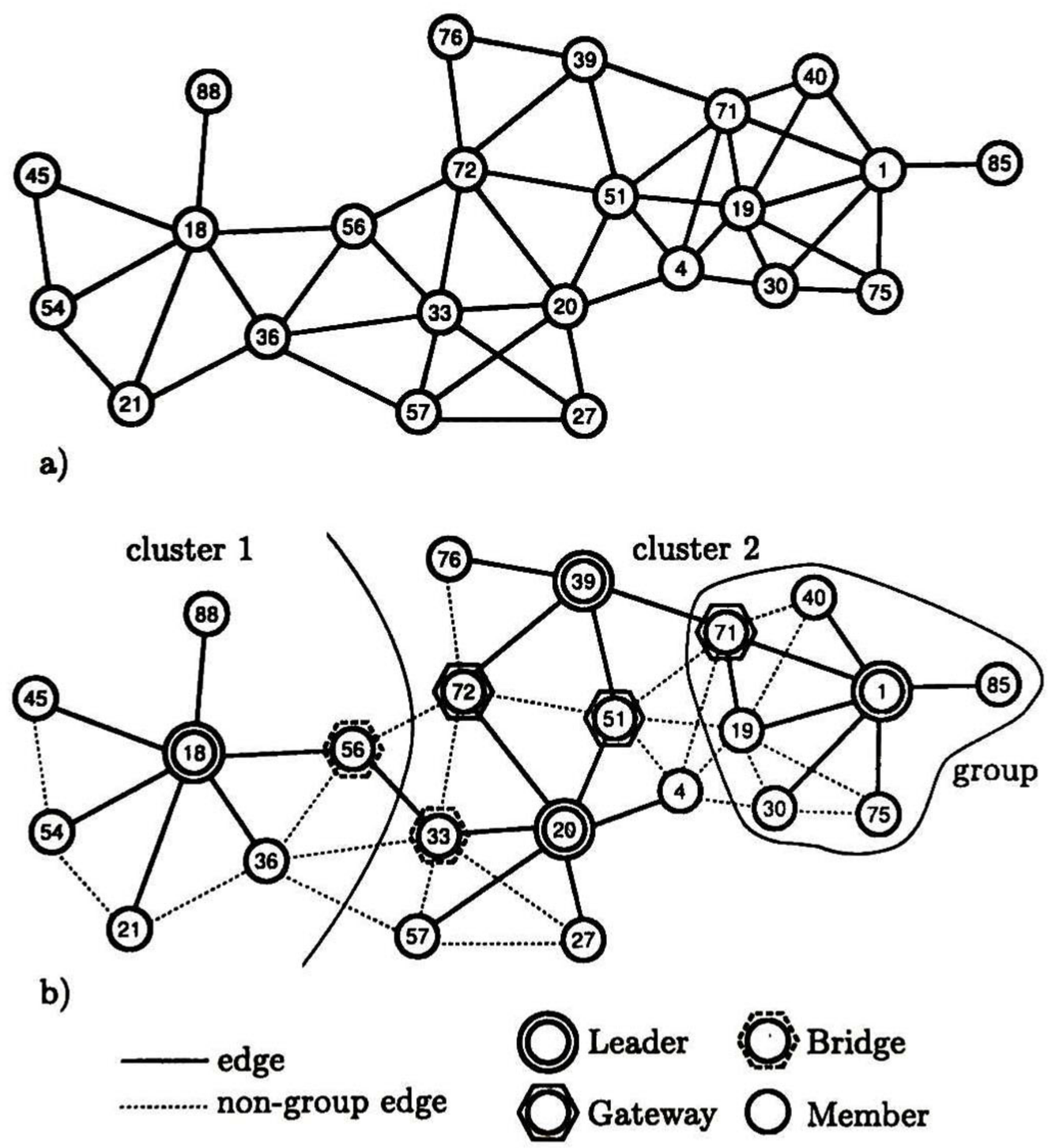

Figure 3.1: Clustering for global approach a) Original graph, b) One possible configuration

3.1.3.2 Network modeling

The vertices in a MIS I are designated as *leaders*. Consequently, for each $v \in I$, $N[v] = N(v) \cup \{v\}$ forms a group. The nodes in the wireless network are weighted appropriately to find a weighted backbone that can yield the more suitable leaders.

3.1.4 Optimization Approach

The proposed approach search a subset of vertices of smallest cardinality, which is both independent and dominating in a graph having a structure sprang off from a wireless communication topology. This subset will construct groups within the network, and a hierarchical view of the network is obtained. To the best of our knowledge, a polynomial time algorithm for this problem is not known in the literature. Many optimization problems, including the Minimum Dominating and Maximum Independent Set problems remain NP-complete even when restricting the input to this class of graphs (23). Given a vertex cover of a graph, all vertices which are not in the cover define an independent set; and because the vertex cover problem is NP-complete (19), then computing the MIS is also NP-complete.

3.1.4.1 Probability for the Random Variables

The random variables g_i and m_i are introduced to describe the event the node v_i takes the role of *gateway* and *member* respectively. The g_i and m_i are defined as Bernoulli random variables. For example:

$$g_i = \begin{cases} 0 & \text{if } v_i \text{ is not a gateway} \\ 1 & \text{if } v_i \text{ is a gateway} \end{cases} \tag{3.1}$$

m_i is defined in the same way. Afterwards, the probability of event v_i to be a gateway or not a gateway is defined as follows:

$$f_g(g_i|v_i) = \begin{cases} 0 & |N_L(v_i)| \in \{0,1\} \\ 1 & |N_L(v_i)| > 1 \end{cases} \tag{3.2}$$

Also, the random variable m_i is defined as a random variable with the following probability distribution:

$$f_m(m_i|v_i) = \begin{cases} 0 & \text{if } |N_L(v_i)| \neq 1 \\ 1 & \text{if } |N_L(v_i)| = 1 \end{cases} \tag{3.3}$$

Selecting a leader is the most important issue in the network formation. The virtual backbone is built by using the nodes with the best capability. Therefore, a weight function needs to be defined.

$$\omega_u = \alpha * |\Gamma(u)| + (1 - \alpha) * |\Gamma(C(u))| \tag{3.4}$$

Where ω_u define the weight of a node u, and $|A|$ is the cardinality of a given set A. The variable $\alpha \in [0, 1]$ is used to express a complementary weight between $N(u)$ and $\Gamma(C(u))$. The next equations compute the average value μ between the $\delta(G)$ and $\Delta(G)$ in order to calculate the probability of a node to be leader.

$$\mu_N = \frac{\delta(G) + \Delta(G)}{2} \tag{3.5}$$

$$\mu_C = \frac{\delta_C(G) + \Delta_C(G)}{2} \tag{3.6}$$

$$\mu = \alpha * \mu_N + (1 - \alpha) * \mu_C \tag{3.7}$$

Then, the probability of a leader is defined by using the logistic distribution:

$$F_l(\omega_u; \mu, s) = Pr(X \leq \omega_u) = \frac{1}{1 + e^{-(\omega_u - \mu)/s}} \tag{3.8}$$

where ω_u is the random variable, μ is the mean, and s is a parameter proportional to the standard deviation.

3.1.4.2 Multi-Objective Problem

Given a set of nodes, $V = \{v_1, v_2, \cdots, v_N\}$ randomly deployed. Our problem has the following characterization: Energy-efficiency is of the utmost importance, as it is closely related to the network lifetime. Furthermore, virtual backbone connectivity ought to be as energy-efficient as possible because of batteries can not be replaced. That is why it is necessary to find a subset $I \subset V$ such that logically the number of:

Obj 1 Inactive links are maximized

Obj 2 Leaders are minimized: this is one of the most important objectives because the leader nodes are those that form the groups.

Obj 3 Radio members to leader are maximized: member nodes are in charge of doing their own task, since the more member nodes are selected, the more energy is saved.

Obj 4 Gateways are minimized: it is only necessary $n-1$ gateways per n leaders.

Obj 5 Clusters are minimized: while less bridge nodes are used, it will be better.

Obj 6 Finally, it maximizes the probability to be leader: it is necessary to build the backbone through the nodes with higher degree.

The subset I is considered as the optimal set to build the virtual backbone, where each $v \in I$ is a leader.

After a MIS has been computed, it is necessary to define which nodes are gateways, members and bridges. Then, in order to describe the Multi-Objective Optimization Problem (MOOP), the following sets are defined:

- The gateway vertices set of G is defined as $V_g = \{v \in V : f_g(g|v) = 1\}$ (equation 3.2).
- The member vertices set of G is defined as $V_m = \{v \in V : f_m(m|v) = 1\}$ (equation 3.3).

Notice that when there is no direct communication between some leader and gateway throughout the network, there are more than one cluster or sub-network (See Figure 3.1). The function $Sub(I)$ provides the number of clusters in the network given a MIS.

Finally, the objective functions are formulated as follows:

$$Max\ f_1 = |E \backslash E'| \tag{3.9}$$

$$Min\ f_2 = |I| \tag{3.10}$$

$$Max\ f_3 = |V_m| \tag{3.11}$$

$$Min\ f_4 = |V_g| \tag{3.12}$$

$$Max\ f_5 = \sum_{u \in I} F_l(u) \tag{3.13}$$

$$Min\ f_6 = Sub(I) \tag{3.14}$$

Subject to:

$$\forall u \in V \backslash I, \exists v \in I : (u, v) \in E \tag{3.15}$$

Therefore, any solution I will be a feasible solution to the problem if I is a MIS, i.e., $I \subseteq V$ satisfies the requirement that each vertex in $V \backslash I$ is neighbor to at least one vertex in I. As stated later, the solution to this MOOP is a set $I \subset V$.

3.1.4.3 Pareto-Optimality

To apply a PSO algorithm in multi-objective optimization, the single-objective scheme has to be modified to cope with the fact that the solution of a problem with multiple objectives is not a single one but a set of non-dominated solutions (7).

In order to handle both minimization and maximization of multiple objective functions, it is used the operator $\preceq$ between two solutions x_1 and x_2 as $x_1 \preceq x_2$ denotes that solution x_1 is better than solution x_2 on a particular objective. Similarly, $x_1 \not\succeq x_2$ for a particular objective implies that solution x_1 is no worse than solution x_2 on this objective (12).

Definition 3.1.1. A solution x_1 dominates other solution x_2 iff both conditions 1 and 2 are true:

1. The solution x_1 is no worse than x_2 in all objectives, or $f_j(x_1) \not\succeq f_j(x_2)$ for all $j = 1, 2, \cdots, M$. Where M is the number of objective functions.

2. The solution x_1 is strictly better than x_2 in at least one objective, or $f_j(x_1) \preceq f_j(x_2)$ for at least one $j = \{1, 2, \cdots, M\}$.

3.1.5 Particle Swarm Optimization

3.1.5.1 Basic PSO Strategy

Traditional PSO algorithms are directly applicable to evolutionary search strategies limited to continuous search spaces problems. Therefore, it is necessary to modify PSO for applications in combinatorial optimization problems of wireless ad-hoc networks.

PSO is in principle a multi-agent parallel search technique, which consists of three steps, namely, generating positions and velocities of particles, velocity update, and position update. Particles are conceptual entities that constitute a swarm, which fly

through the multidimensional search space. It finds the global optimum by simply adjusting the trajectory of each individual towards its own best location and towards the best particle of the swarm at each generation of evolution (37).

In order to update the velocity v_i, and position of the particle x_i, the next equations are used in conventional PSO algorithms:

$$v_i(t+1) = wv_i(t) \quad + \quad c_1R_1\left[pBest_i(t) - x_i(t)\right] + c_2R_2\left[gBest_i(t) - x_i(t)\right]$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{3.16}$$

where w is the inertia weight; $i = 1, 2, \cdots, q$ indicates the number of particles in the population; $t = 1, 2, \cdots, t_{max}$, indicates the generations (iterations). $v_i = [v_{i1}, v_{i2}, \cdots, v_{in}]^T$ is the velocity of the ith particle; $pBest_i$ and $gBest_i$ represents the best previous position of the ith particle and global best position in the swarm respectively.

Positive constants c_1 and c_2 are the cognitive and social factors respectively, which are the acceleration constants responsible for varying the particle velocity towards $pBest$ and $gBest$, respectively. Variables R_1 and R_2 are two random variables with uniform distribution in the range $[0, 1]$.

3.1.5.2 The discrete PSO

Now, a new PSO algorithm is proposed; it determines an optimal virtual backbone given a graph of a wireless network to improve the network lifetime.

First of all, the parameters are initialized; a random position and velocity are given to every particle. The position of a particle is encoded as a binary string $x_i(t) = (x_{i1}, x_{i2}, \cdots, x_{in})$ where x_{ij} is 1 if the jth node in the particle i is selected as leader, otherwise it will be 0.

A recursive algorithm is used to generate q possible solutions for a set of N nodes. In order to ensure a complete exploration a MIS drawing upon each vertex is generated.

Algorithm 1 receives two nodes (initially u_p and v represent the same node, and then u_p will be the previous visited node), and puts a solution in I that satisfy the constrains for the desired structure.

Algorithm 4 calculateMIS(u_p, v)

Require: $v \in G(V, E)$

Ensure: $MIS = I$

```
I ← I ∪ {v}
N_v ← N_v ∪ {u_p}
while ∃u ∈ N(v) : u ∉ {I ∪ N_v(v)} do
    N_v(v) ← N_v(v) ∪ {u}
    while ∃w ∈ N(u) : w ∉ N[I] do
        calculateMIS(u, w)
    end while
end while
```

The velocity is represented as $v_i(t) = (v_{i1}, v_{i2}, \cdots, v_{in})$ where v_{ij} are randomly initialized following a uniform distribution in the range of $[0, 1]$.

A problem that arises when using the global best position is that after a number of iterations, the swarm collapses due to complete diversity loss. This implies that further exploration is not possible and the particles can perform only local search around their convergence point. This point possibly lies in the neighborhood of the overall best position (37).

The former problem is addressed by introducing the concept of neighborhood. The main idea is to reduce the global information exchange scheme for a more local one, where information is diffused only in a reduced number of swarm particles at each iteration. More precisely, each particle assumes a set of other particles to be its neighbors, and at each iteration, it communicates its best position only to these particles, instead of to the whole swarm. This strategy emulates those used in random field operators as the Gibbs sampler and the Metropolis-Hastings sampler (40).

This can be expressed formally. Let p_i be the ith particle of a swarm $S = \{p_1, p_2, \ldots, p_q\}$. Then, a neighborhood of p_i is defined as $NB_i = \{p_{n_1}, p_{n_2}, \cdots, p_{n_r}\}$, where $\{n_1, n_2, \cdots, n_r\} \subseteq \{1, 2, \cdots, q\}$ is the set of indices of its neighbors. The topology used for the formation of neighborhoods is based on particle indices. Thus, the neighborhood of p_i can be defined as: $NB_i = \{p_{i-r}, p_{i-r-1}, \cdots, p_{i-1}, p_i, p_{i-1}, \cdots, p_{i+r-1}, p_{i+r}\}$; the neighborhood radius r will depend on the number of vertices in G.

In this case, R_1 and R_2 should be considered as random N-dimensional vectors with their components uniformly distributed within $[0, 1]$, and c_1 and c_2 are eliminated. The

inertia weight w shall be selected such that the effect of $v_i(t)$ fades during the execution of the PSO algorithm. Thus, a decreasing value of w along time is preferable. This scheme for w can be mathematically described as follows:

$$w(t) = w_{up} - (w_{up} - w_{low})\frac{t}{T_{max}} \tag{3.17}$$

where t stands for the iteration counter; w_{low} and w_{up} are the desirable lower and upper bounds of w; and T_{max} is the total allowed number of iterations (37).

Finding the non-dominated set from a given set of solutions is similar in principle to finding the minimum of a set of real numbers. The dominance relation $\preceq$ defined previously can be used to identify the best of two given solutions.

A weighted density estimator De is used to sort the set of non-dominated solutions.

$$De = \gamma_1 * f_1 - \gamma_2 * f_2 + \gamma_3 * f_3 - \gamma_4 * f_4 + \gamma_5 * f_5 - \gamma_6 * f_6, \tag{3.18}$$

where $\sum_{i=1}^{6} \gamma_i = 1$.

The strategy used to find the non-dominated set is the efficient method by Kung et al. (12). This method works as follows:

Step 1 Sort the population according to the descending order of importance in the density estimator (De) and rename the population as P of size N.

Step 2 **Front(P)** If $|P| = 1$, return P as the output of **Front(P)**. Otherwise, $T = Front(P^{(1)} - P^{(|P|/2)})$ and $B = Front(P^{(|P|/2+1)} - P^{(|P|)})$. If the ith non-dominated solution of B is not dominated by any non-dominated solution of T, create a merged set $P_{non} = T \cup \{i\}$. Return P_{non} as the output of **Front(P)**.

What finally returns from Step 2 is the non-dominated set.

3.1.5.3 PSO for the addressed problem

Each particle can store the best position it has ever visited during its search. but the best position is not a single value. Actually, it is a non-dominated set. Given a particle p_i, every new position x_i the particle finds is stored into its own non-dominated set NDs_i, and then the Front($\cdot$) method is applied.

At each iteration, after the best personal positions are updated, the local best positions in the neighborhood are computed; $LNDs_i = \cup_{j \in NB_i} NDs_j$, then the Front($\cdot$) method is applied also to eliminate the dominated positions.

Considering the previous modifications, (3.16) is rewritten as:

$$v_i(t+1) = wv_i(t) \quad + \quad R_1\left[pBest_i(t) - x_i(t)\right] + R_2\left[lBest_i(t) - x_i(t)\right] \tag{3.19}$$

where $pBest_i(t) = NDs_{ij}$, and $lBest_i = LNDs_{ij}$; the j index is chosen randomly, in order to ensure a further exploration. During the iterations, the particle will be updated and evaluated repeatedly until the maximum iteration.

Now a movement through the search space by specifying the equations of motion must be performed. $R_1\left[pBest_i(t) - x_i(t)\right]$ gives a velocity as well as $R_2\left[lBest_i(t) - x_i(t)\right]$, where the subtraction operation $(-)$ between $xBest_i$ and x_i position is computed by the following equation:

$$xBest_{ij} - x_{ij} = \begin{cases} 0 & \text{if } x_{ij} = 1 \\ xBest_{ij} & \text{otherwise} \end{cases} \tag{3.20}$$

In this regard, the next step is to add each velocity term in equation 3.19. In order to ensure the values of the velocity are in the range of $[0, 1]$ it is necessary to introduce a mechanism in such a way that the accumulated velocity of each position j of the velocity v_i is further bounded by means of the following constraint:

$$v_{ij}(t+1) = \begin{cases} 1 & \text{if } v_{ij}(t+1) > 1 \\ v_{ij}(t+1) & \text{otherwise} \end{cases} \tag{3.21}$$

Once the velocity has been computed, the position of the particle is calculated. In order to do the sum $(+)$ operation between $x_i(t)$ and $v_i(t+1)$ (see equation 3.16), a position $j \in \{k : v_{ik} = 1\}$ is randomly selected. Then, the j position in x_i will switch to 1. In addition, a mechanism is introduced such that the new position x of each particle p is further bounded by means of position constrains. To this end, an interesting property has been found in the MIS problem such that it can be modified without loss of independence.

Definition 3.1.2. Given sets $A = \{x | x \textit{ represents a } MIS\}$, $B = \{1, 2, ..., N\}$ and $A' = \{x' | x' \textit{ represents a } MIS\}$. The function $\varphi : A \times B \rightarrow A'$ is defined as $\varphi(x_i, x_{ij}) = x'_i$ which is computed as follows.

1. $x_{ij} = 1$
2. $\forall x_{ik} \in N_L(x_{ij})$, $x_{ik} = 0$.
3. $\forall x_{il} \in N(x_{ik})$, if $|N_L(x_{il})| = 0$ then $x_{il} = 1$.

Proposition 1. Given a particle x_i such that the set of leaders forms a MIS in G and an index that represents a particular vertex in x_i. If function $\varphi(x_i, x_{ij})$ is applied to any $x_i \in A$, then the result x'_i represents a MIS.

Proof. First, we will name x_i as x'_i after the function φ is performed. Now, the graph $G(V, E)$, where leaders are x'_i, is split in three different subgraphs:

1. $X = G(N[x_{ij}], E_1)$ where $E_1 \subseteq E$ are the edges between the nodes in $N(x_{ij})$.
2. $Y = G(V_Y = \{x_{il} | x_{il} \in N(x_{ik}) \wedge x_{ik} \in N_L(x_{ij})\} \setminus V_X, E_2)$ where $E_2 \subseteq E$ are the edges between the nodes in V_Y.
3. $Z = (V \setminus \{V_X \cup V_Y\}, E \setminus \{E_1 \bigcup E_2\})$.

The changes produced by the application of (1), (2) and (3) in definition 3.1.2 can only affect no more than 2-hops, i.e., the leader set in the neighbors of x_{ij} and x_{ik}. Thus, we only need to prove that in each subgraph X, Y and Z, their sets of leaders form a MIS. And from this, when we consider the union of the sets of vertices and the set of edges of the graphs X, Y and Z, the union of the sets of leaders forms a MIS in G, i.e., the leaders in x'_i forms a MIS in G. To accomplish this, we will prove that for each subgraph X, Y their set of leaders form an independent set, which is a MIS.

First, starting with X, let suppose the set of leaders $X_l \subseteq X$ is not an independent set in X. Then, there must be two vertices $x, y \in X_l$ representing an edge, but $|X_l| = 1$ because of operation (2). Therefore, X_l is an independent set and maximal. Now, let us suppose the set of leaders Y_l of Y does not form an independent set. Thus, there must be two vertices $x, y \in Y_l$ such that $(x, y) \in Y(E)$. This is not possible because of operation (3). Then, Y_l is an independent set and maximal also. Now, $Z_l \subseteq Z$ is an independent set because of the construction of Z. In addition, the leaders in Z are an independent set because they were part of the MIS formed by x_i before conditions were applied, and the way operation (3) was applied.

Finally, when taking into account the edges among the graphs (see Figure 3.2):

1. The vertices in the boundary of X are non-leaders.

2. The vertices between Y and Z are leaders or non-leaders because of (3). Thus, two leaders do not share the same edge.

Therefore, the set of leaders in x_i' forms an independent set in G. We have to determine if this set is maximal: assume that there exists a set M of leaders such that $X_l \cup Y_l \cup Z_l \subset M$. Then, there exists a vertex y that is a leader such that $y \notin X_l \cup Y_l \cup Z_l$. Clearly, $y \notin Z$ because of the structure of Z. Then, y there must be in X or in Y. If $y \in X$, then $y = x_{ij}$ because the only leader in X is x_{ij}. Thus, $y \in Y$, but again by (3), if $y \notin Y_l$ then there are another leader vertex y' such that $(y, y') \in E$, a contradiction. Thus, $X_l \cup Y_l \cup Z_l$ is a MIS. □

Therefore, by using function φ, it is possible to flip elements in the particles to leader role, and as long as we take care of keeping the conditions (1),(2) and (3), the proposition 1 hold, and the new position of particle is still a MIS.

Figure 3.2 shows the switching of a leader node. Currently in Figure 3.2a nodes 7, 3, 1, 10 and 14 are selected as leaders. Let us suppose that node 0 is elected to be a leader. Afterwards, nodes 1 and 7 must change their role in order to preserve the independence property. At the same time the node 5 takes the role of leader because it does not have any leader node as neighbor.

The procedure for implementing the PSO algorithm using multi-objective criteria is given below:

Step 1 Initialize a population compound of q particles with random positions and velocities in the N dimensional problem space using the Algorithm 4. Define the maximum generation T_{max}, and set the generation counter, $t = 0$.

Step 2 Evaluate the particles and store the non-dominated particles in the set *pBest*.

Step 3 Compute the density estimator values of each non-dominated solution in the set *pBest*, and then sort the non-dominated solutions based on this value in descending order.

Step 4 Delete the dominated solutions by using the **Front**$(\cdot)$ function.

Step 5 Get the neighborhood *lBest* of the particle given the value of r, and apply the step 4.

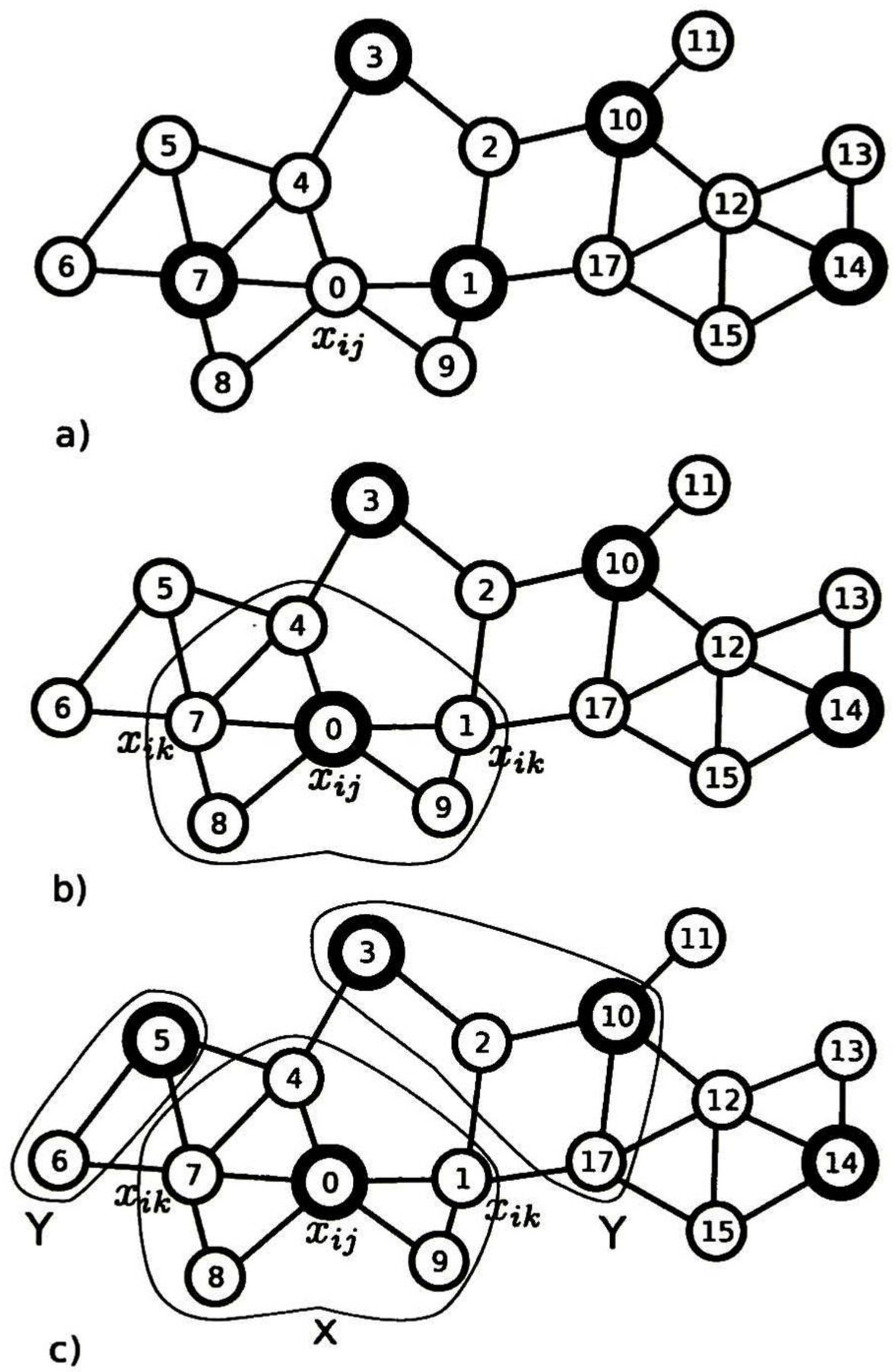

Figure 3.2: Switching from x_i to x_i' - a) current position of particle x_i. b) x_{ij} turns to leader. c) new position of particle x_i.

Step 6 Select randomly from the set *pBest* a personal best position as well as the local best from *lBest*.

Step 7 Evaluate (3.19) for obtaining the velocity v_i for the generation $t+1$.

Step 8 Change the position of the particle, x_i, according to the mutation operation in (3.16).

Step 9 Increment the generation counter t in a unit.

Step 10 Return to Step 2 until the T_{max} value is reached.

3.1.5.4 Example

Let x_1 be the position of the particle p_1 and its velocity v_1 in a certain time t. The current position of x_1 is showed in Figure 3.3a.

$$x_1(t) = (1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 1)$$

$$v_1(t) = (0.5\ 0\ 0\ 0.7\ 0.2\ 0.6\ 0\ 0.3\ 0.7\ 0\ 0.2\ 0.4\ 0.5\ 0.8\ 0\ 0.7\ 0.6\ 0.4\ 0\ 0\ 0.8\ 0\ 0.1)$$

Assume that its personal best *pBest* and local best *lBest* of x_1 is as follows:

$$pBest_1(t) = (1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0)$$

$$lBest_1(t) = (0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1)$$

After the subtraction between the best positions $xBest_1$ and x_1 the result is:

$$pBest_1(t) - x_1(t) = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0)$$

$$lBest_1(t) - x_1(t) = (0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0)$$

When the multiplication is applied the result is a velocity.

$$R_1\,(pBest_1(t) - x_1(t)) = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.5\ 0\ 0.3\ 0\ 0.1\ 0\ 0)$$

$$R_2\,(lBest_1(t) - x_1(t)) = (0\ 0.3\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0.5\ 0\ 0\ 0.2\ 0\ 0\ 0\ 0.6\ 0\ 0\ 0\ 0)$$

As result of the sum of the three computed velocities, the velocity $v_1(t+1)$ is obtained.

$$v_1(t+1) = (0.5\ 0.3\ 0\ 0.7\ 0.2\ 0.6\ 0\ 0.3\ 0.7\ 0\ 0.2\ 0.9\ 0.5\ 0.8\ 0.2\ 0.7\ 1\ 0.4\ 0.9\ 0\ 0.9\ 0\ 0.1)$$

Finally, the updating is done and the $x_1(t+1)$ is obtained (see Figure 3.3b).

$$x_1(t+1) = (1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1)$$

3.1.6 Simulation and Performance Assessment

This section discusses the simulation results and the performance of the proposed algorithm. It is assumed that there are 100 nodes randomly deployed in a plane. Figure 3.4 shows two different virtual backbones obtained from simulations in a region of 100 times 100 meters. The simulations in Figure 3.4 only illustrate how the virtual backbone is formed. However, it is of utmost importance to validate the uniformity of the Pareto front set. To this end, we must be able to assess the performance respect to some performance indicators. The main bulk of the proposed algorithms in literature are not validated regarding this performance indicators. With the aim to assess the performance of results Diversity-Based indicators are used (7).

Clearly, a well-diversified solution set with a uniform distribution over the entire range of feasible domain is desired. Figure 3.5 shows the set of non-dominated solution points obtained from the environment previously mentioned. Where the solutions set is showed by three of the six different functions.

A space whose coordinates are the objective functions is usually referred as the objective space, in which, each feasible solution point corresponds to one coordinate. Pareto solution set needs to be validated, i.e., it is necessary to know that it is a well-diversified solution set with a uniform distribution over the entire range of the objective space. If the solutions are grouped into clusters, there are several solution points in each cluster that are not significantly different from one to another and there are areas in the domain with few or no solution points.

In order to measure the combined uniformity and coverage of a Pareto set, and considering the location and distribution of all solution points, a simple metric based on the concept of entropy introduced by Ali Farhang-Mehr (17) is used. Shannon's entropy measures how evenly a set of numbers is spread. If the values of the entries in this

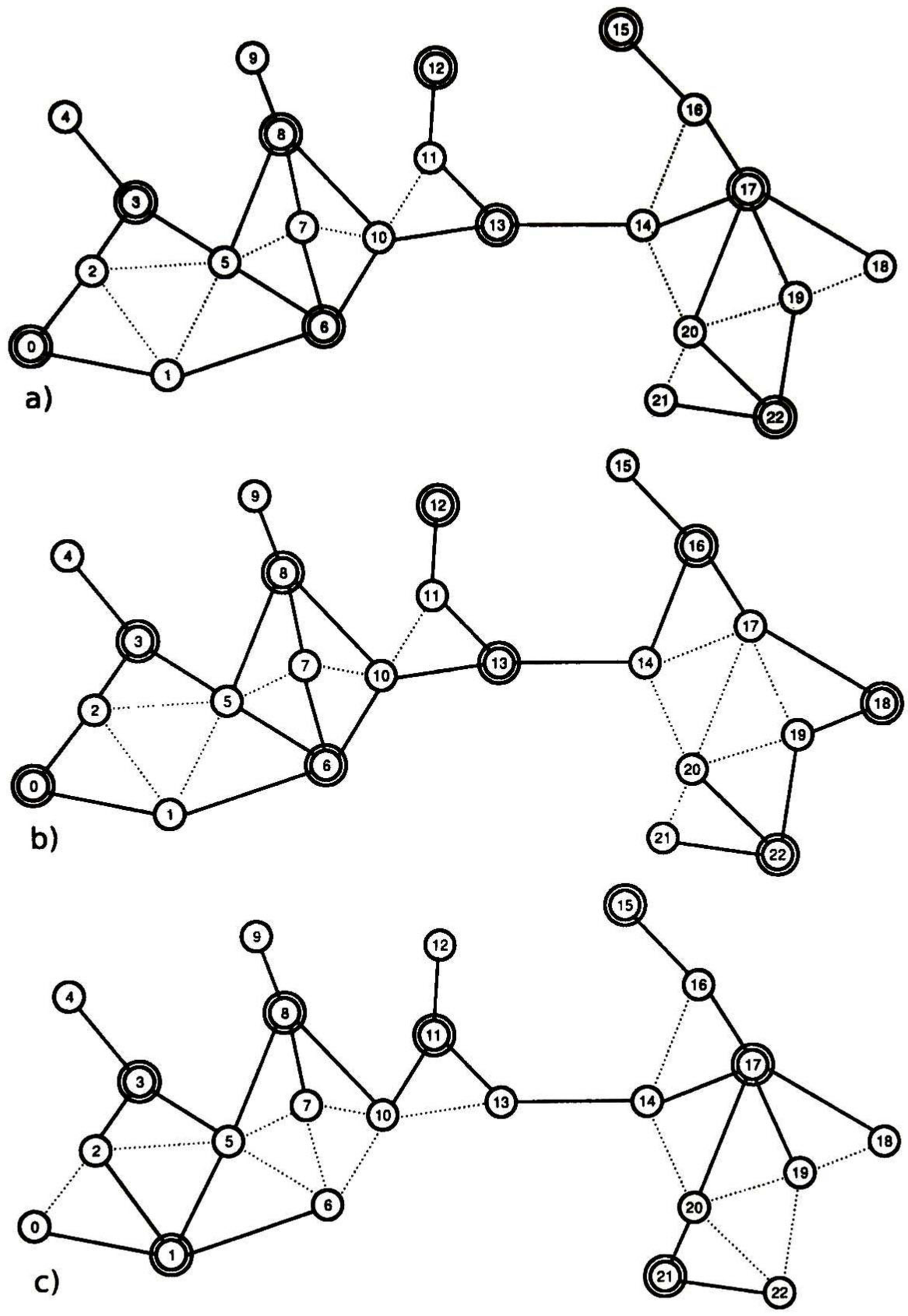

Figure 3.3: Particle Movement - a) particle x_i at time t. b) Particle x_i at time $t+1$, c) The optimum position of particle x_i.

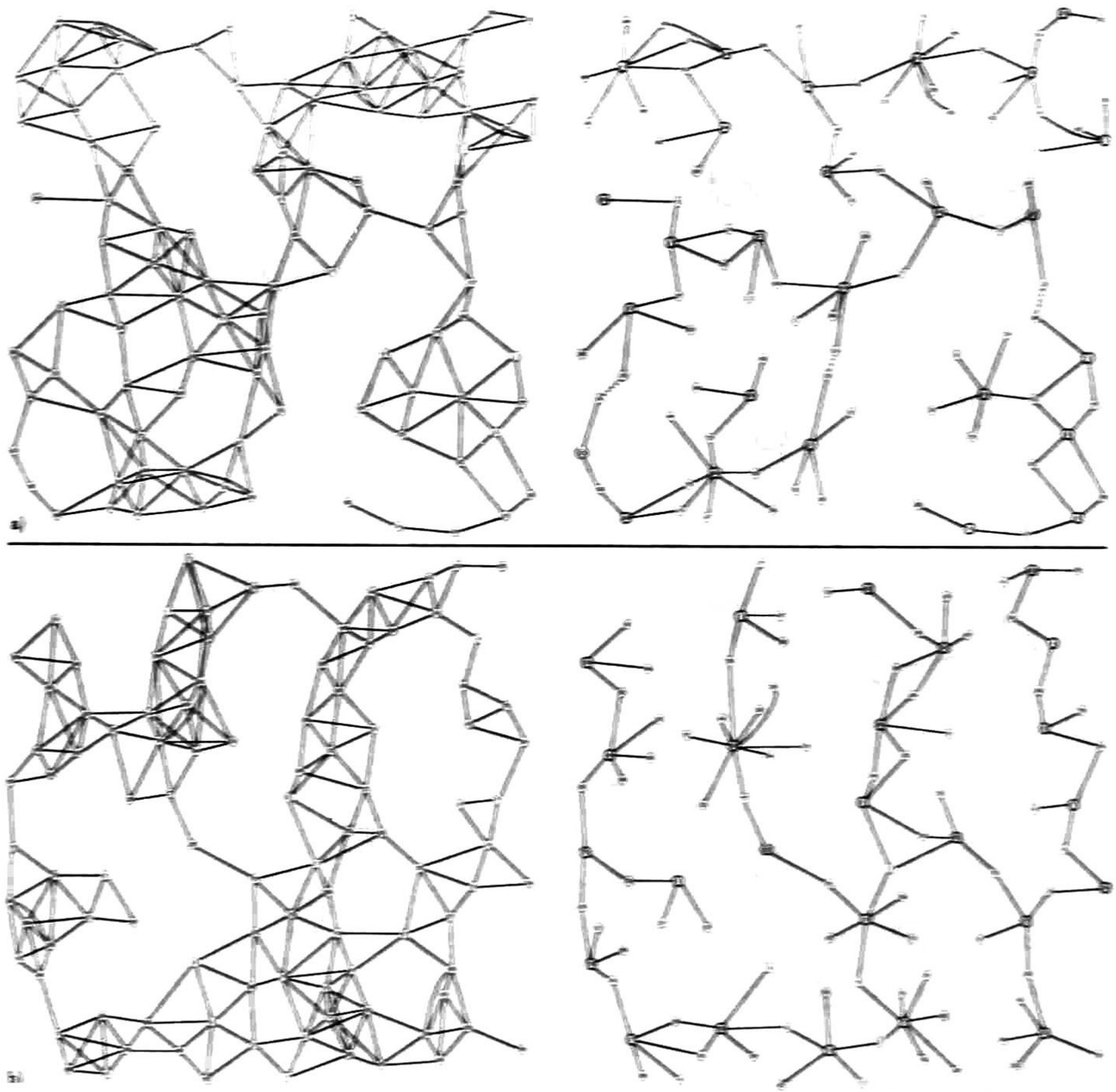

Figure 3.4: Global results - Virtual backbones obtained from different environments

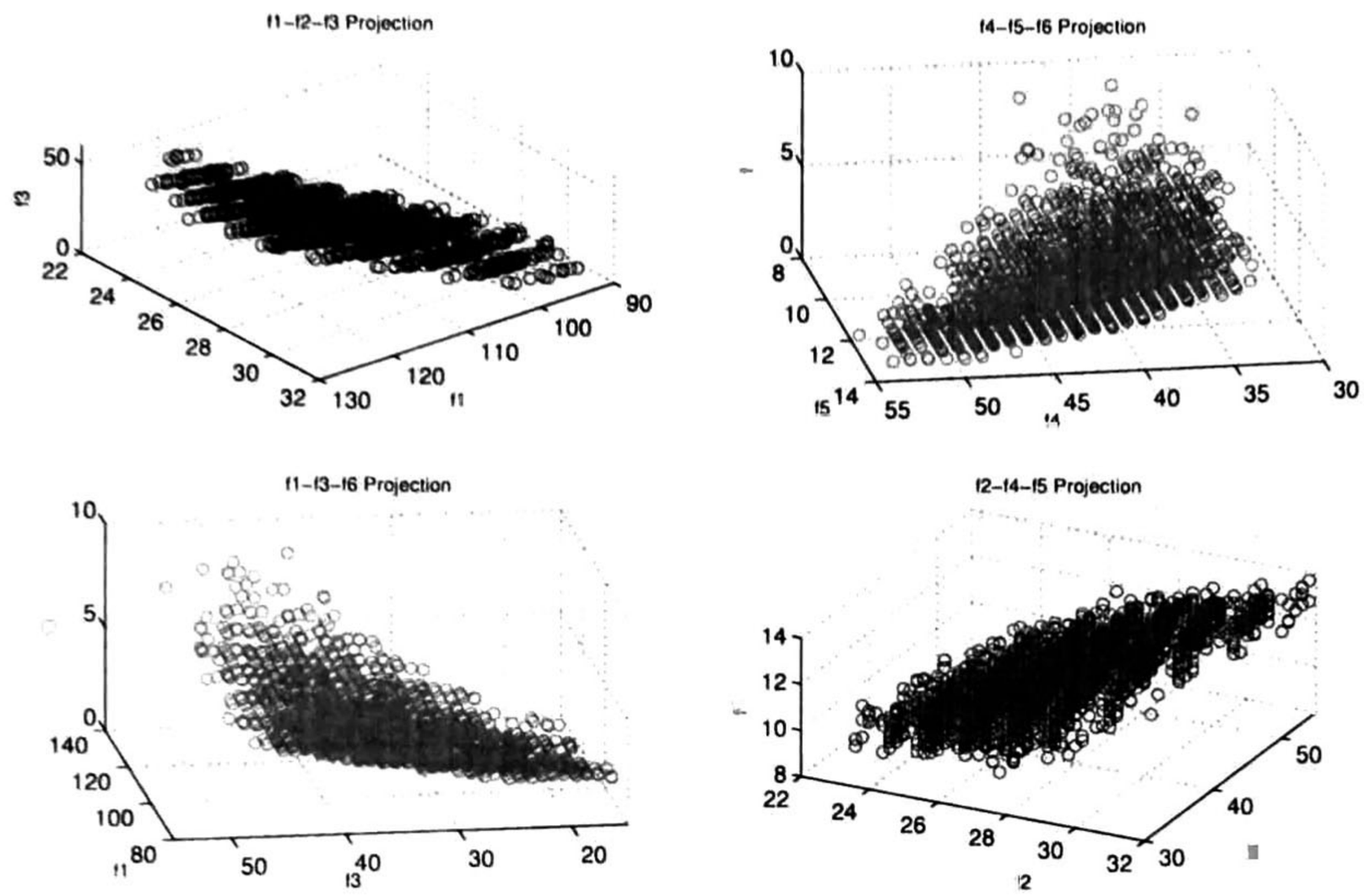

Figure 3.5: Pareto frontier - Projection of three different functions

vector are approximately the same then the entropy is large, but if the entries are very different, the corresponding entropy is low. In order to take advantage of the entropy the influence and density function need to be used. Informally, the influence function is a mathematical description of the influence a point has within its neighborhood, and the density function is defined as the sum of the influence functions of all the points (22). In principle, the influence function can be an arbitrary function. For simplicity, the Gaussian influence function is selected:

$$f(x,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}} \tag{3.22}$$

Then, the density function which results from a Gaussian influence function is:

$$f_D(x) = \sum_{y=1}^{N} e^{-\frac{d(x,y)^2}{2\sigma^2}} \tag{3.23}$$

where $d(x,y)$ represents the Euclidean distance function. In order to evaluate the entropy, the feasible domain is divided into $a_1 \times a_2 \cdots \times a_6$ grid cells. Then the density function at each cell $f_{D_{ijklmn}}$ is obtained from (3.23).

Table 3.1: Diversity-Based Indicator

Environment		Virtual Backbone				Entropy	
# Nodes	Dim.	L	G	M	B	H	H_{max}
14	20x20	3	2	9	0	0.5426	0.6931
23	30x30	7	5	9	2	1.8065	2.3026
50	50x50	11	9	26	4	3.7860	4.8675
100	100x100	25	32	44	0	4.1174	5.2281
200	150x150	57	65	70	8	6.9070	9.0646

Finally, for the 6-dimensional objective space, the entropy is defined as follows:

$$H = -\sum_{i=1}^{a_1}\sum_{j=1}^{a_2}\cdots\sum_{n=1}^{a_6}\rho_{ijklmn}ln(\rho_{ijklmn}) \tag{3.24}$$

where ρ_{ijklmn} is the normalized density. The result is a single entropy scalar that evaluates the quality of a set of alternatives. In simple words, a solution set with higher entropy is spread more evenly throughout the feasible region, and provides a better coverage of the space (17). Here, the maximum entropy is $H_{max} = ln(n)$, where n is the number of grid cells in the search space.

The performance has been evaluated through several simulations using diverse scenarios. Table 3.1 summarizes some simulations. It is clear that results are good enough regarding the properties of uniformity and space coverage.

3.1.7 Conclusion

In this chapter, a novel algorithm is introduced to efficiently generate an energy-efficient communication strategy for low-resource, large-scale wireless ad-hoc networks. This strategy states the problem as a multi-objective optimization on a graph; then it uses a particle swarm optimization algorithm to obtain an optimal solution. Furthermore, we have created a method for calculating an optimal virtual backbone among the nodes taking advantage of the independence property. The multi-objective approach allows representing the different functional constraints under the problem of virtual backbone generation. The simulations show that this approach is efficient and effective to obtain a well spread solutions in the Pareto frontier. However, further research needs to be done to compare this algorithm with the more traditional methods.

Chapter 4

Experiments and Discussions

Abstract. This chapter presents simulations of MWAC-based and EESOA algorithms using NS-2. Energy consumption and network topology analysis is carried out. Furthermore, the effect of agents mobility on the network structure is discussed and an approach for solving this problem is stated.

4.1 Network Simulator

The behavior and performance of the proposed algorithms have been analyzed via simulation using NS-2 version 2.33 (11), an event oriented simulator for network research. In real networks, obstacles can interfere the signal and cause random anisotropic signal strengths. The topology of an ad-hoc network depends on several uncontrollable factors such as agent mobility and interference.

The radio propagation models implemented in NS-2 are used to predict the received signal power of each packet. There is a receiving threshold at the physical layer of each wireless agent. When a packet is received, if its signal power is below the receiving threshold, it is marked as error and dropped by the MAC layer.

The shadowing model extends the ideal circle model to a richer statistic model: agents can only communicate probabilistically when they are near the limit of the communication range, i.e., the communication range in the shadowing model is not an

ideal circle. For this, an inverse Q-function is used to calculate the receiving threshold. This section presents a sample of simulation tests.

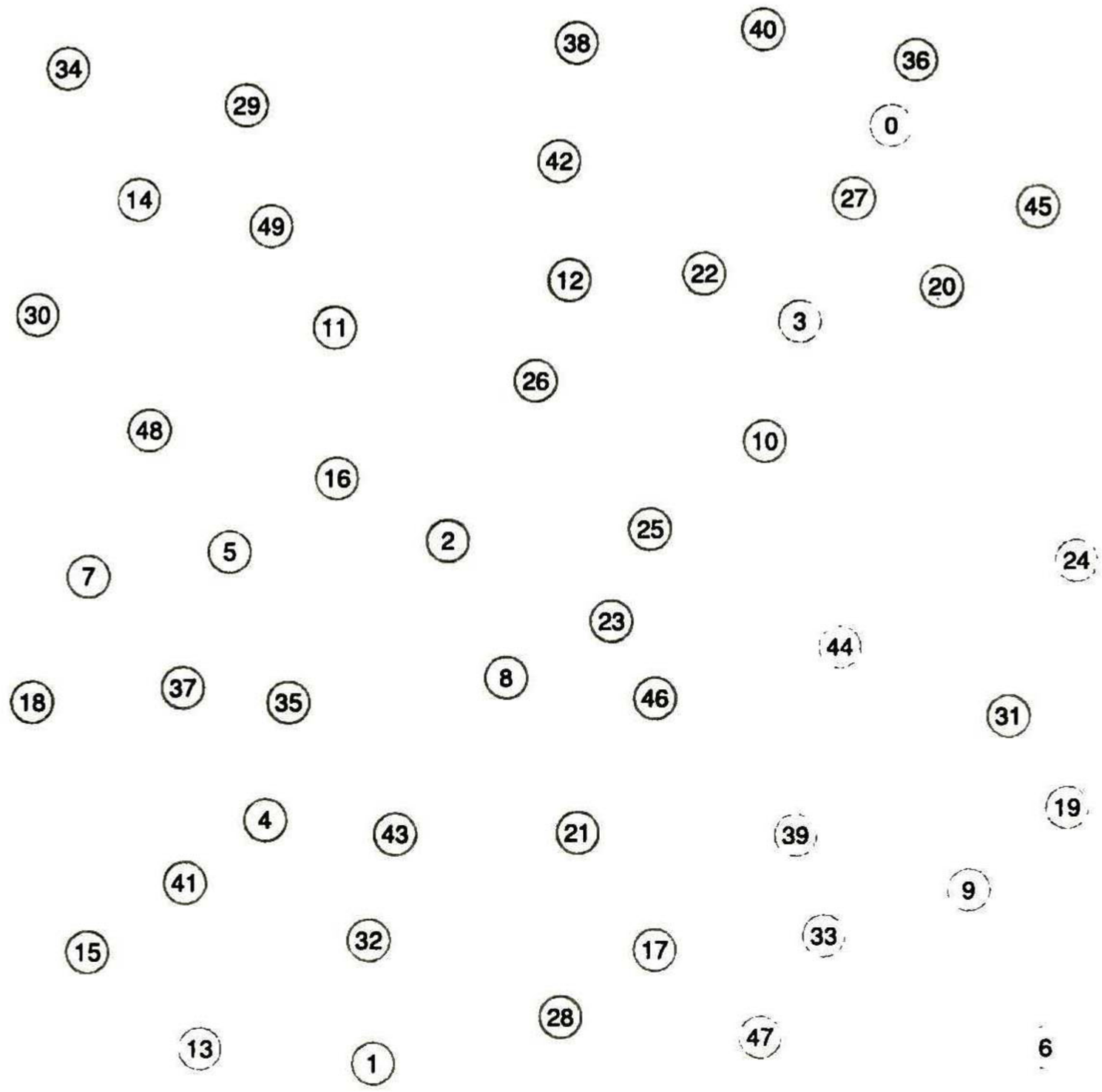

Figure 4.1: 50-Agents scenario - Initial environment of 50 agents randomly deployed.

4.2 MWAC-based Simulations

4.2.1 Scenario

Consider a scenario where a wireless sensor network has to be organized. It consists of 50 mobile agents distributed within an environment of 50 times 50 meters (Figure 4.1). For simplicity assume that the agents have the same constraints (although this is not strictly required), and they are turned on simultaneously. The initial configuration of agents is:

The network interface is 802.15.4.

The initial energy of every agent is 2 joules.

- The tasks devoted to receive messages, transmit messages, and mode change (active-slept) consume energy.
- The maximum transmission range is 10 meters.

Two versions of the proposed algorithm were tested:

- One considers static transmission power (MWAC).
- The other one considers variable transmission power and variable transmission period.

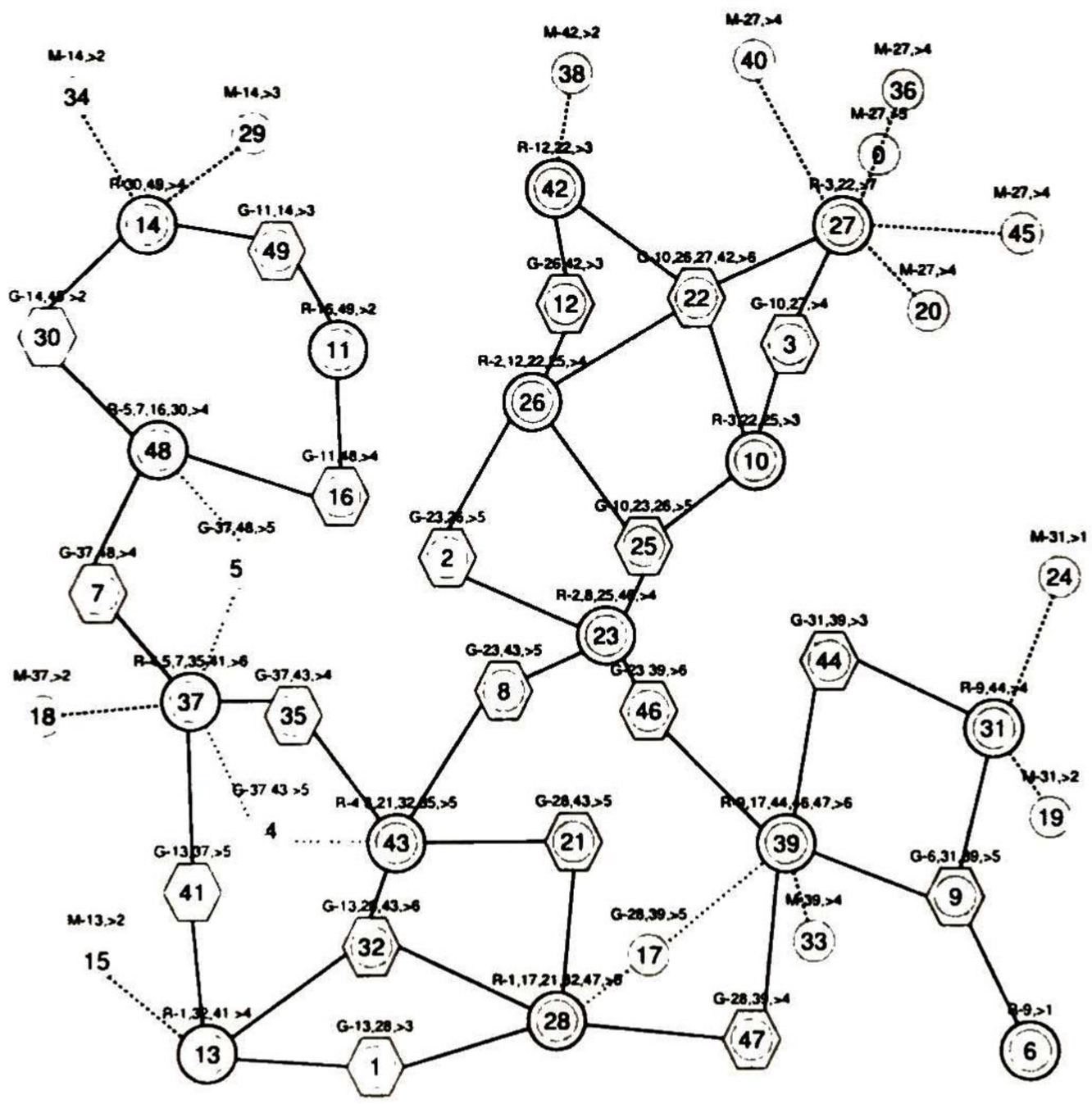

Figure 4.2: Virtual backbone of 50-agent scenario Self-Organization Strategy (MWAC-Based).

4.2.1.1 Running NS-2

For comparison purposes a basic algorithm MWAC (26) which is representative of related works has been implemented. The main features of such algorithm are: localized, distributed, and emergent behavior.

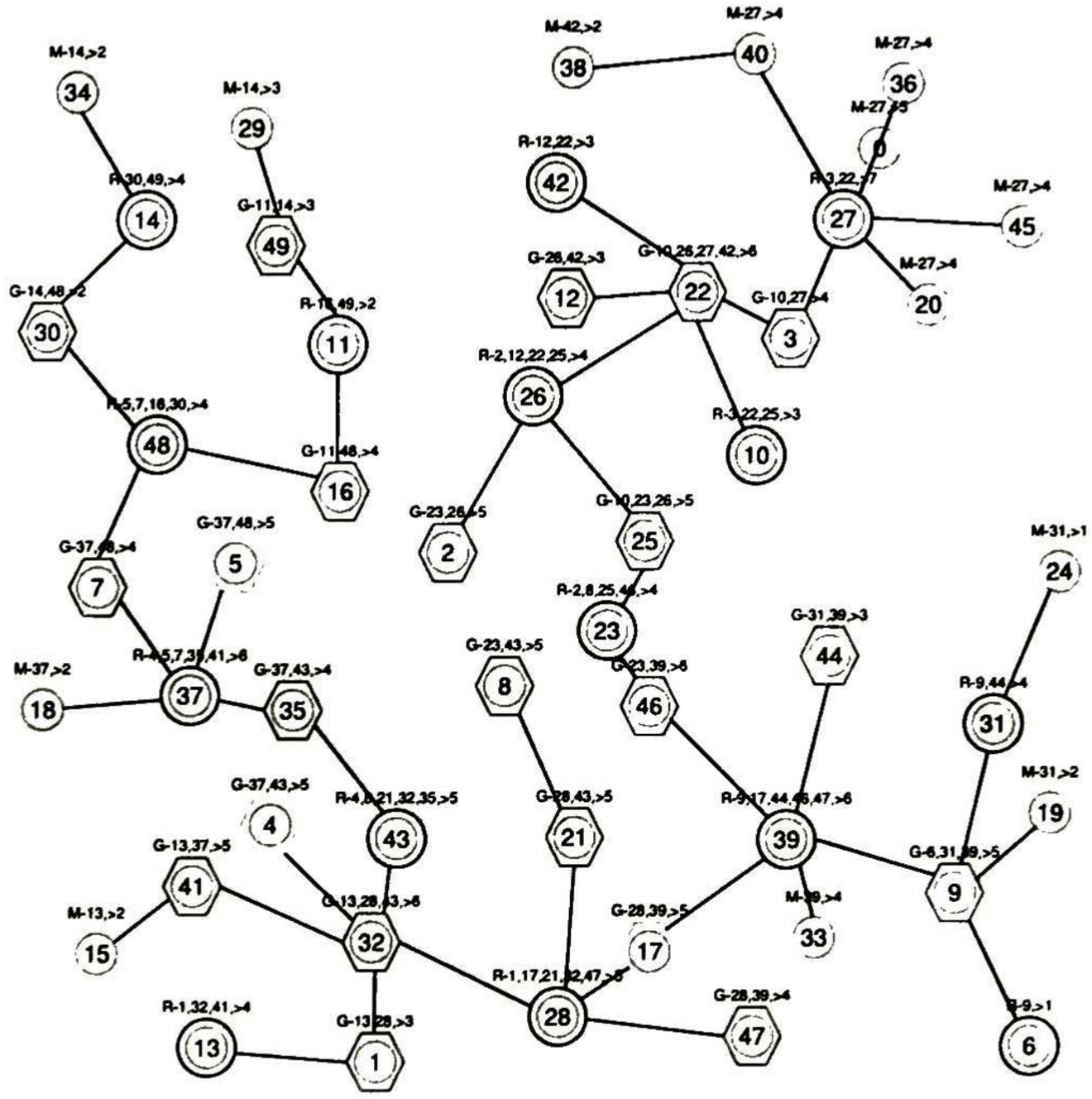

Figure 4.3: MST of 50-agent scenario - Minimum Spanning Tree

The simulation in Figure 4.2 shows the backbone generated at the instant $t = 14$ when the backbone connects the whole network. This behavior is obtained using both versions of the algorithm. In environments where agents are turned on at the same time, the network can be created in segments because the agents interact only with their neighbors. This problem is fixed by bridge agents. However, it is not possible to predict how the network will behave during the structuring. Even more, when an agent is removed, the restructuring process may generate a different virtual backbone. In the situation where the agents are turned on simultaneously, a conflict appears when the agents detect neighbors, and decide to take the role of leader. This will start a

negotiation process between the agents in conflict yielding high energy consumption. When the agents turn on asynchronously, the network is built gradually yielding a structure without problems.

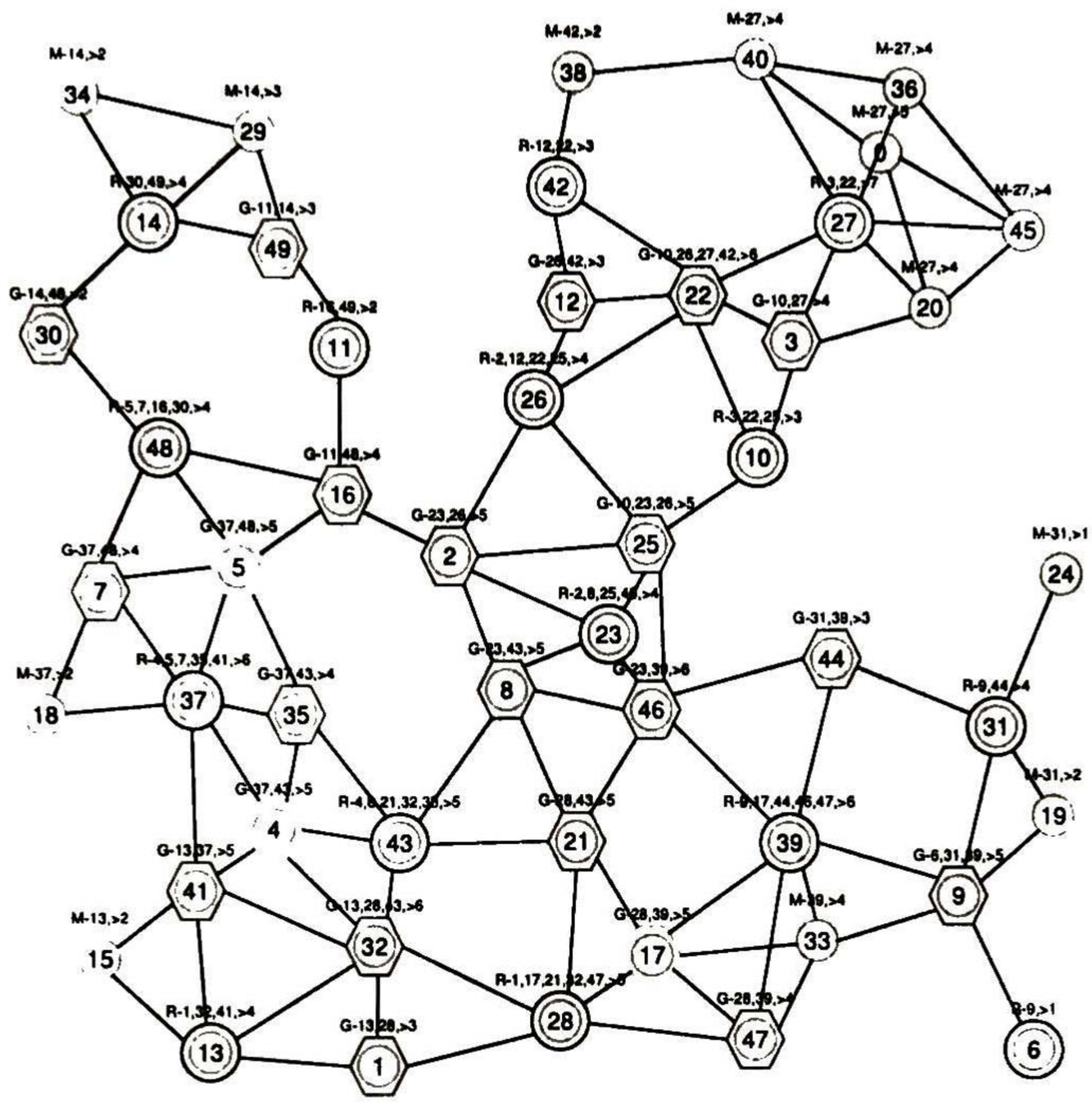

Figure 4.4: UDG of 50-agent scenario - Unit Disk Graph

In Figure 4.2 we have approximately 56 links connecting the entire network without taking into account the gateway agents in sleeping state. When we construct the minimum spanning tree (Figure 4.3) in the initial environment, we obtain a result similar to the proposed algorithm. The MST is created with 49 links among the agents. Both agents, leader and gateway are in the virtual backbone.

The *Unit Disk Graph (UDG)* $G(V, E)$ is typically used to model ad-hoc wireless networks under this situation (39), where two agents are connected when their distance is within this fixed transmission range. Figure 4.4 shows the graph known as the UDG, where all the agents are homogeneous, containing an edge (u, v) if and only if the unit disks around u and v intersect. It's clear that, there is a great difference between UDG

and the simulation results, where there exist 97 links in UDG against 49 in the proposed algorithm.

It is shown in Figures 4.3-4.4 that our simulation is close to the MST algorithm based on the number of links. In this way, we are close to the optimum using a localized strategy (i.e., the agents do not have the knowledge of the whole environment, they only are aware of their neighborhood) without using extra control messages.

4.2.2 Analyzing Energy Consumption

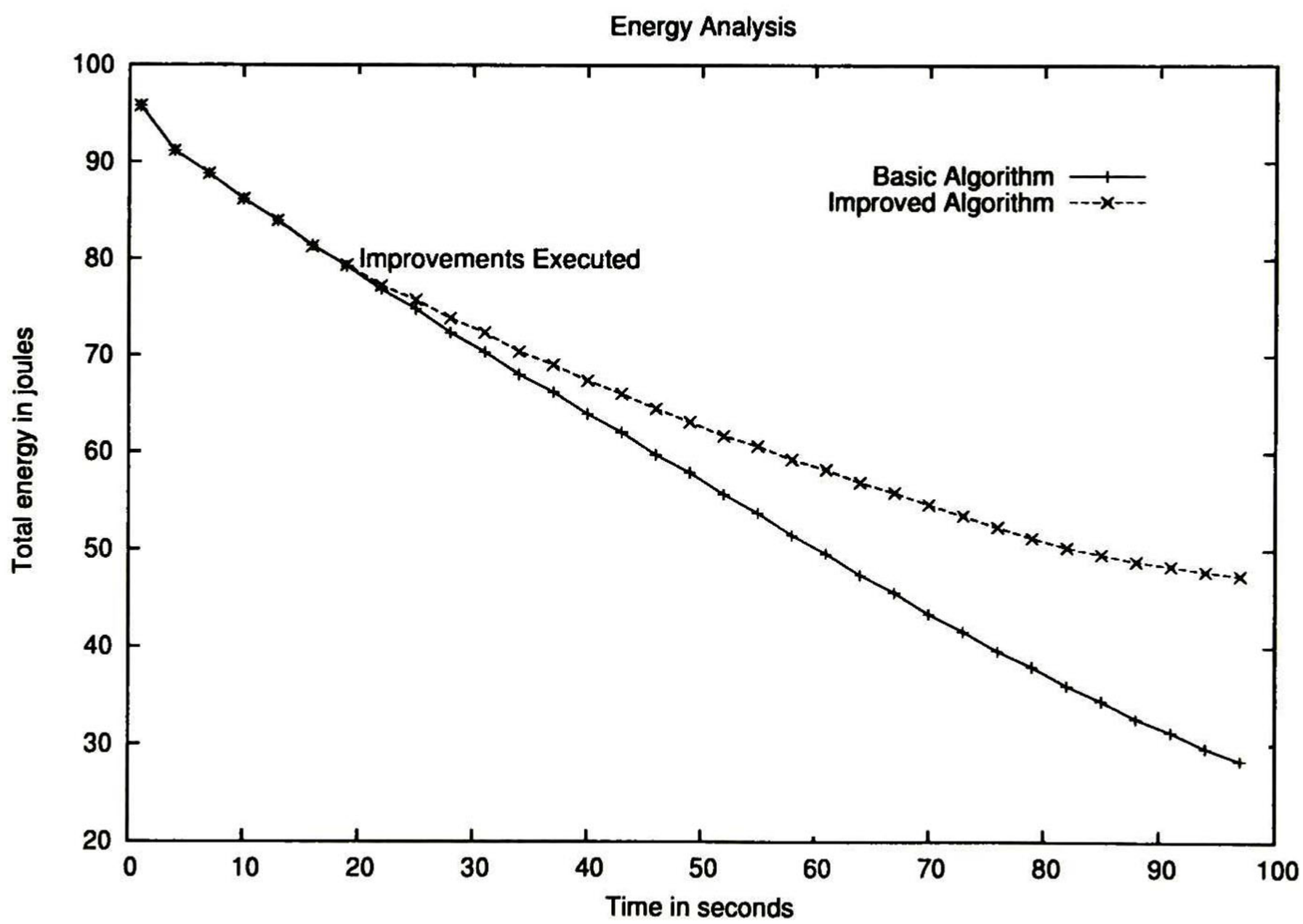

Figure 4.5: Remaining Energy - Improvement of the total energy in the network

The total energy usage is defined as the total sum of each individual energy usage of each agent in the interval $[0, \tau]$, where τ is the maximum simulation time.

Now we are going to compare the energy consumption in the entire network using the two versions of the protocol. The simulation time was $t = 100sec$. It was clear that in real applications this time range is too short, but it is used for comparison purposes. The remaining energy is showed in Figure 4.5. It can be noticed that in the instant

Table 4.1: Energy comparative table

Range	Agents	Dimensions	Initial Energy	Energy Saving
10 Mts	50	50x50	2 Joules	22%
15 Mts	60	100x100	2 Joules	25%
15 Mts	70	100x100	2 Joules	26%
15 Mts	80	100x100	2 Joules	24%
15 Mts	100	100x100	3 Joules	30%

$t = 15$ the consumed energy is equal, since it is the stabilization time of the network. This instant depends on the application. After this time, the curves in the graphic differ from each other. The lower curve corresponds to the energy consumption using the simplest algorithm. The upper curve shows the behavior of the energy consumption when the transmission power of the leader agent is adjusted. In this curve the energy savings at instant $t = 100$ is about 20%. Additionally, at this time t the connectivity is preserved.

This performance has been demonstrated through several simulations using diverse scenarios. Table 4.1 summarize some of these simulations. It is clear that energy saving is better in our strategy. It can be noticed that energy saving is about 20 and 30 percent in comparison to the basic algorithm.

4.2.3 Solving the Segmentation Problem

In order to evaluate the performance of our algorithm under different parameters, we randomly deployed 60 agents in a fixed area of 100×100 meters. The initial transmission range of each agent is 15 meters (Figure 4.6).

Figure 4.7 illustrates the simulation results without using bridge agents. We can see that the member agents 2 and 42 can see each other, but they don't have any connection. This kind of organization gives us network segmentation, which justifies the necessity of bridge agents.

Figures [4.8-4.10] compares the three connected networks in terms of the number of connections. It can be seen in Figure 4.8 that the minimum spanning tree is composed of 59 links among the entire network. The simulation using bridge agents is showed in

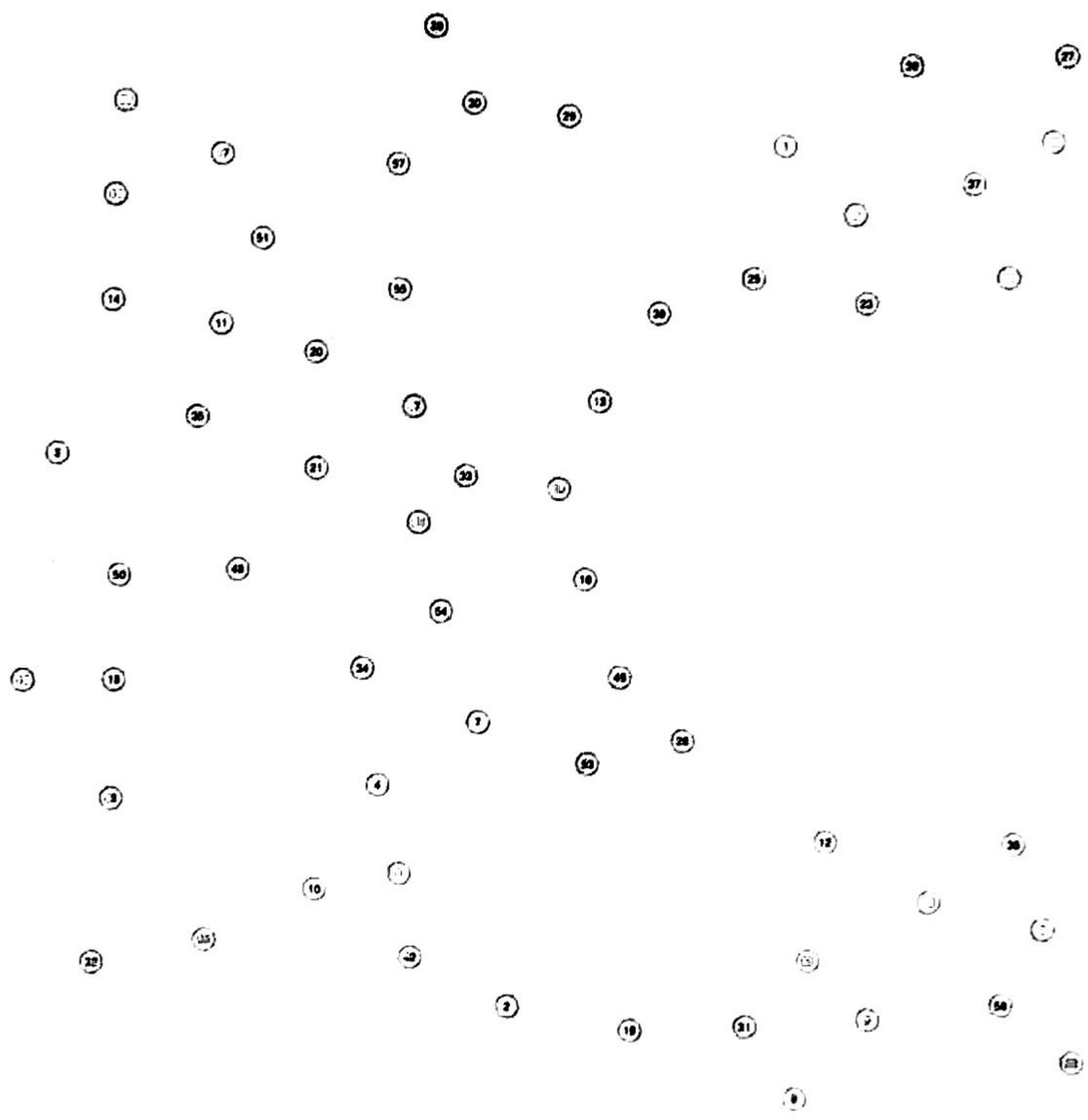

Figure 4.6: 60-Agent scenario - Initial environment of 60 agents randomly deployed.

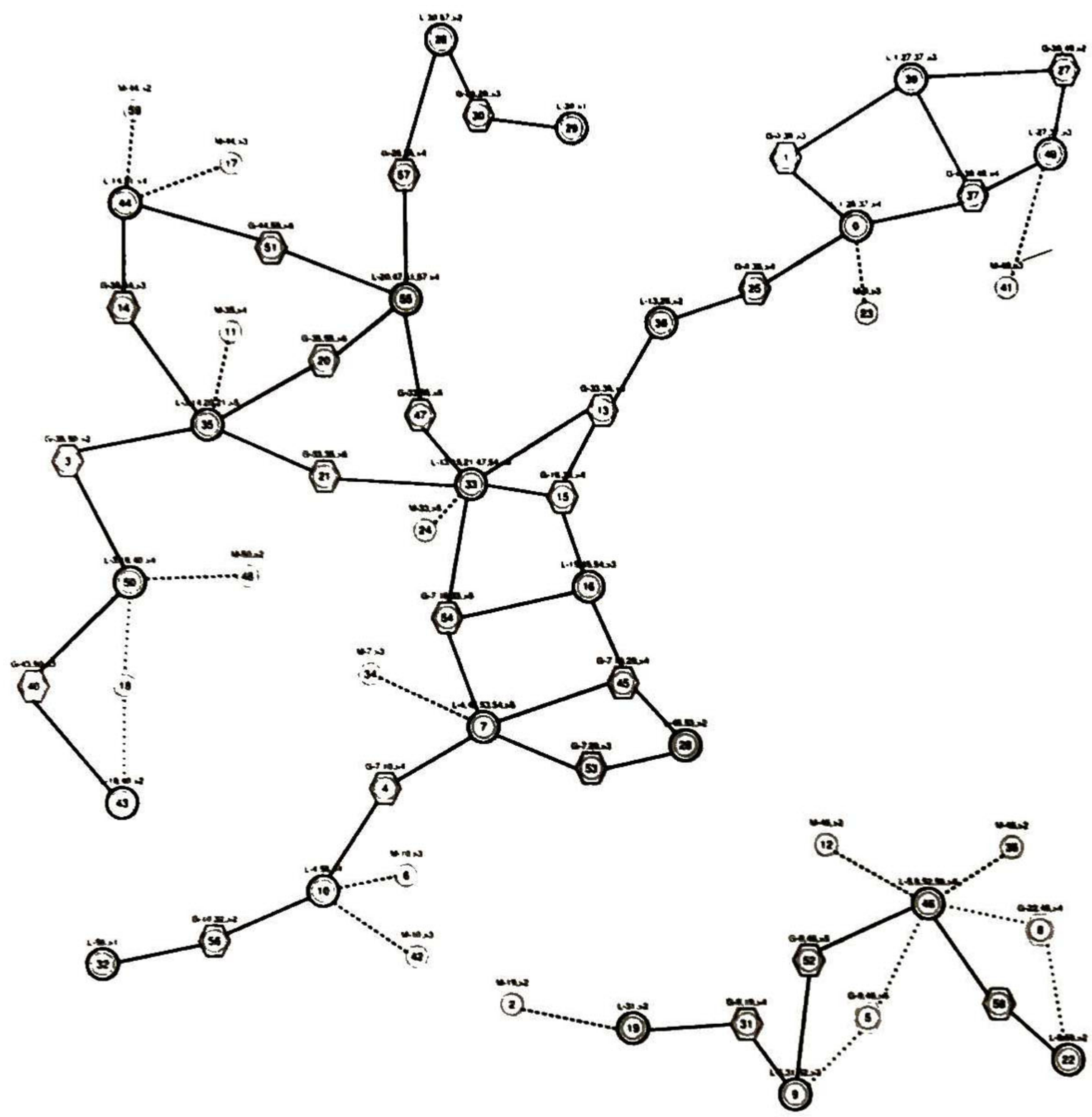

Figure 4.7: Segmented-backbone of 60-agent scenario - Simulation showing segmentation

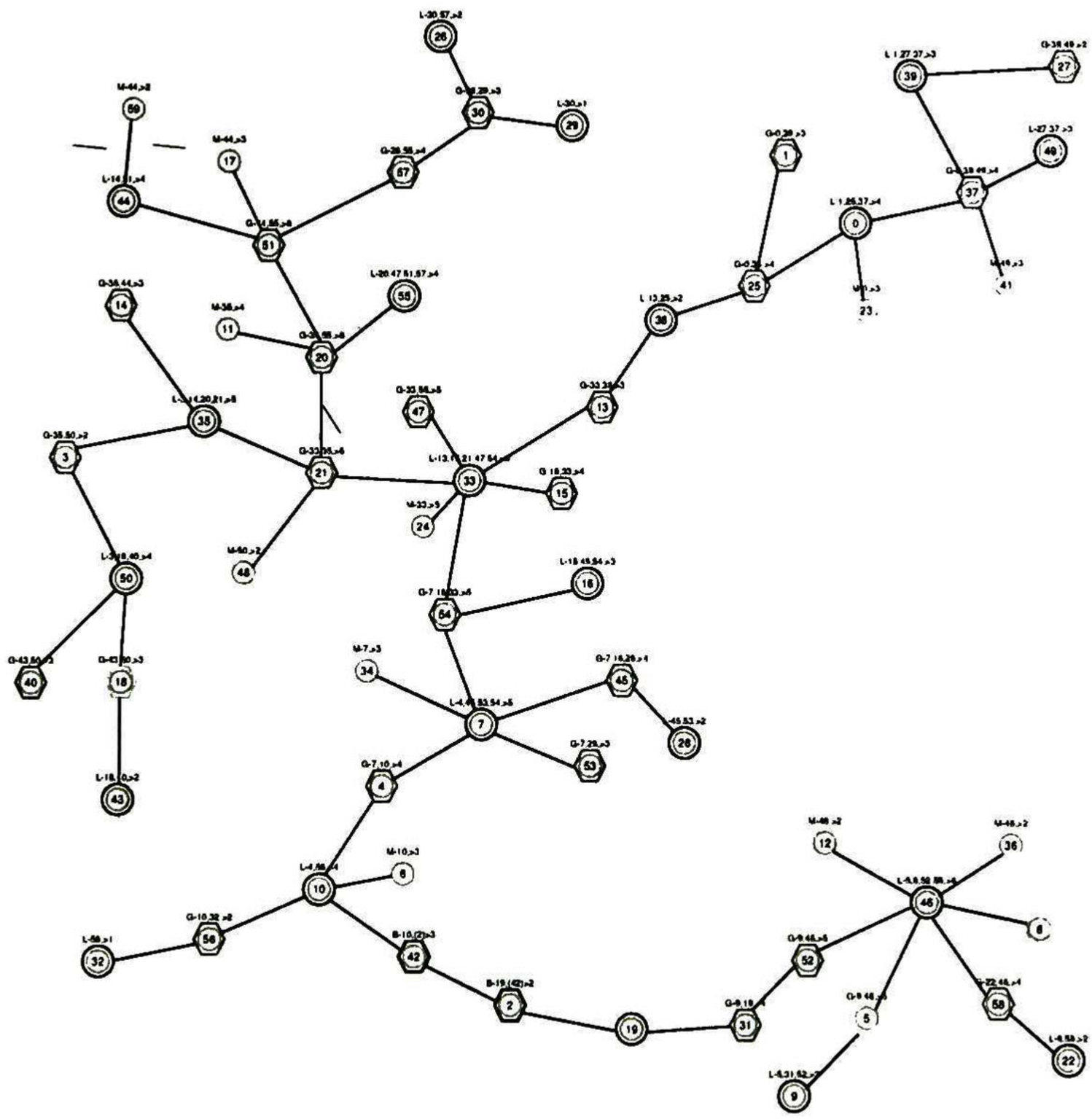

Figure 4.8: MST of 60-agent scenario - Minimum spanning tree

the Figure 4.9. This simulation without segmentation computes 68 links. Specifically, the number of links obtained from the simulation has 9 links more than the minimum spanning tree. This is due to the fact that the communication is localized, this means that the information the agents know is local. As expected, the results indicate that the simulation is near to the minimum spanning tree. In addition, the unit disk graph showed in Figure 4.10 uses 101 links. Note that this number of links is larger than the simulation results.

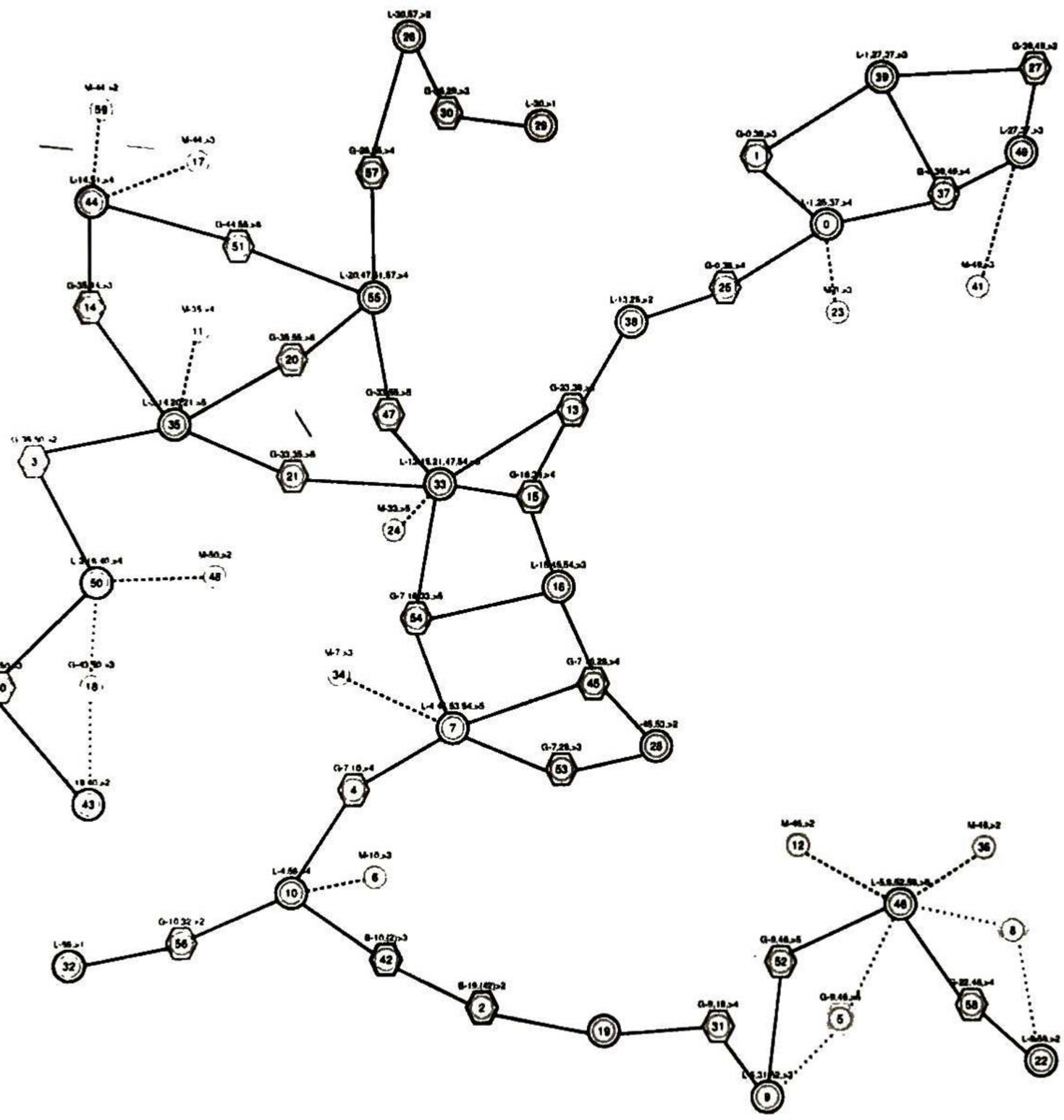

Figure 4.9: Virtual backbone of 60-agent scenario - Segmentation solution using bridge agents

In conclusion, the simulation reveals that these results are consistent with our preliminary analysis. This means that we save energy without the loss of connectivity, and the segmentation problem of the basic approach is alleviated.

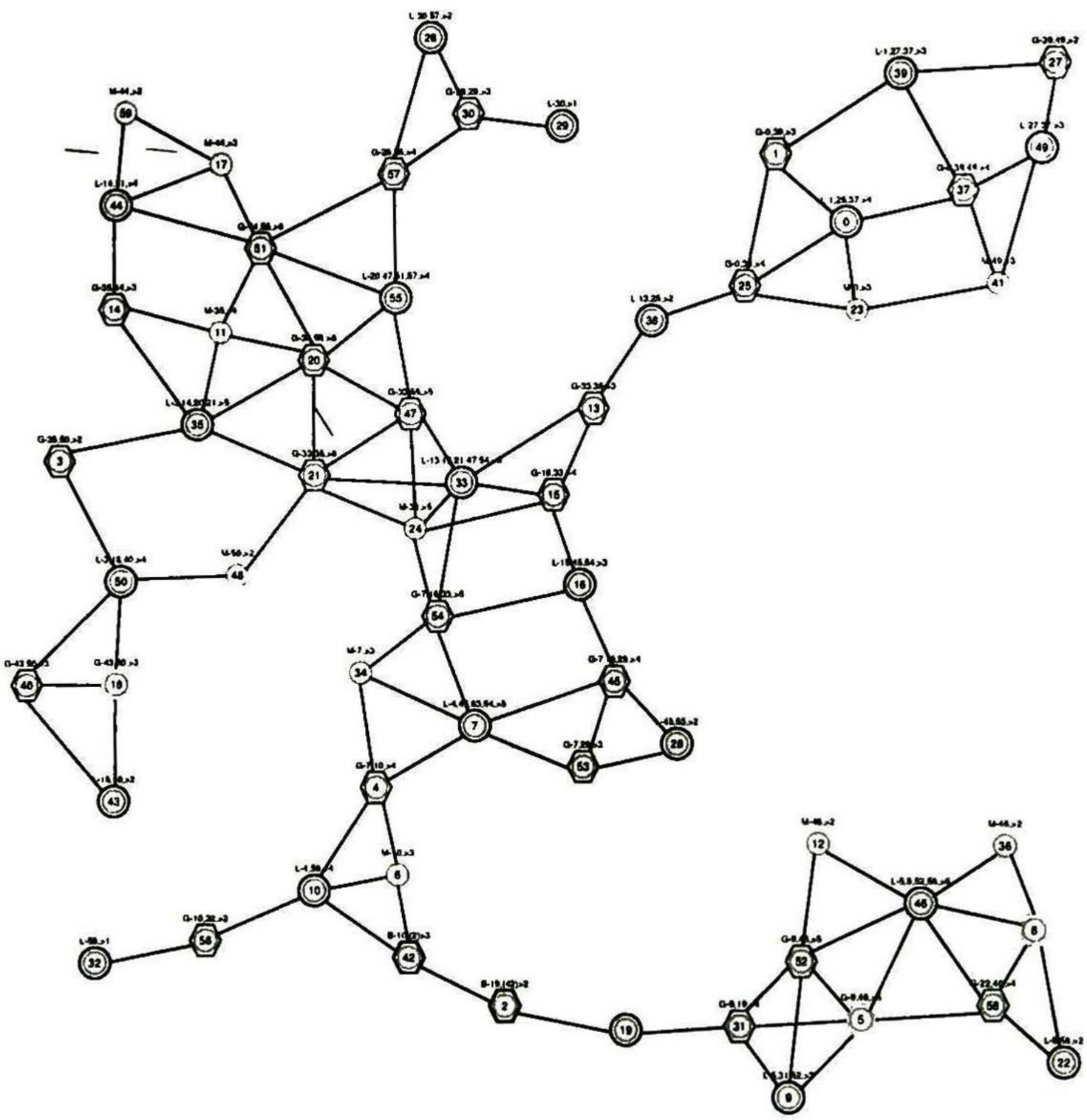

Figure 4.10: UDG of 60-agent scenario - Unit Disk Graph

4.2.4 Approach for Solving Mobility

In this section we conduct the simulation experiments to evaluate the performance in terms of the network's mobility. It is also discussed how agent mobility affects topology control in general, and how our strategy behaves when agents change their positions. The impact of mobility on topology control is twofold:

- Increased message overhead. The implementation of any distributed topology control algorithm causes a certain message overhead which is due to the fact that agents need to exchange messages in order to reorganize the network at once. In the presence of mobility, the topology control algorithm must be executed frequently to take into account the new information of the agents. Thus, reducing message overhead is fundamental when implementing topology control mechanisms in mobile networks.

- Nonuniform agents spatial distribution. Some mobility patterns cause a nonuniform agent spatial distribution. This fact should be carefully taken into account in setting important network parameters (e.g., the ideal transmitting range) at the design stage. From this discussion, it is clear that the impact of mobility on the effectiveness of topology control techniques heavily depends on the mobility pattern. However, the proposed strategy copes with nonuniform agent spatial distribution.

When a member agent moves to another place within the network area, it does not cause any problem to the network because the member agent does not maintain any communication with another agent, just with its own leader. Although a gateway or bridge agent makes communication through the network, the problem is the same as the member. The problem arises when a leader agent moves to another place because of it manages the communication among its member and gateway agents. Even though, the real problem is when the leader agent is moving and transfer the leadership to another agent.

In order to solve this, the only assumption we are doing is that the agent is aware of its movement: under this constraint, when the velocity of an agent is faster than a threshold δ, this agent will abandon its role, and it will not take any role until it reduces its velocity as far as δ in order to avoid interfering another agents in the network.

4.2.4.1 Dynamic Scenario

In order to evaluate the performance of the algorithm, we randomly deployed n agents on a fixed area of 100×100 meters. Consider an environment where a wireless sensor network has to be organized. It consists of 100 mobile agents.

For simplicity we assume that the agents have the same constraints, (although this is not strictly required) and they are turned on simultaneously. The initial conditions of agents are:

- The network interface is 802.15.4
- The initial energy of every agent is 3 joules.
- The maximum transmission range is 15 meters.

As can be seen in Figure 4.11, the network is totally connected i.e., the constructed virtual backbone reach all the agents into the network. If we had not used the bridge role, we would have obtained a segmented network. Notice that the couple of agents $7, 65$ and $40, 31$ are connecting the entire network.

Figure 4.11 also shows how some gateway agents are saving energy. In this case, more than 10% of all agents are in sleep mode to save energy. The gateway agents in sleep mode are $75, 33, 57, 9, 19, 48, 49, 54, 29, 51$ and 12.

4.2.4.2 Moving Agents

In order to evaluate the performance of the strategy when the agents move within the network area, three leader agents are displaced.

The first agent to move is number 15. It starts the movement between the time 14 and 20. Figure 4.12a illustrates a snapshot of movement. After the agent 15 moves, a conflict arises between the agents 13 and 94 because both of them try to be leaders. Figure 4.12b shows that the conflict is solved, since agent 13 won the leadership in the group. In addition, Figure 4.12b also shows when agent 15 stops, it takes the role of leader due to there is not any other leader. After agent 15 takes its role, agents $2, 7, 26$ and 65 join it and become gateway agents.

In the same way, when agent 86 moves towards the agent 16, it leaves its group. This group try to reorganize in order maintain network connectivity. The agent that

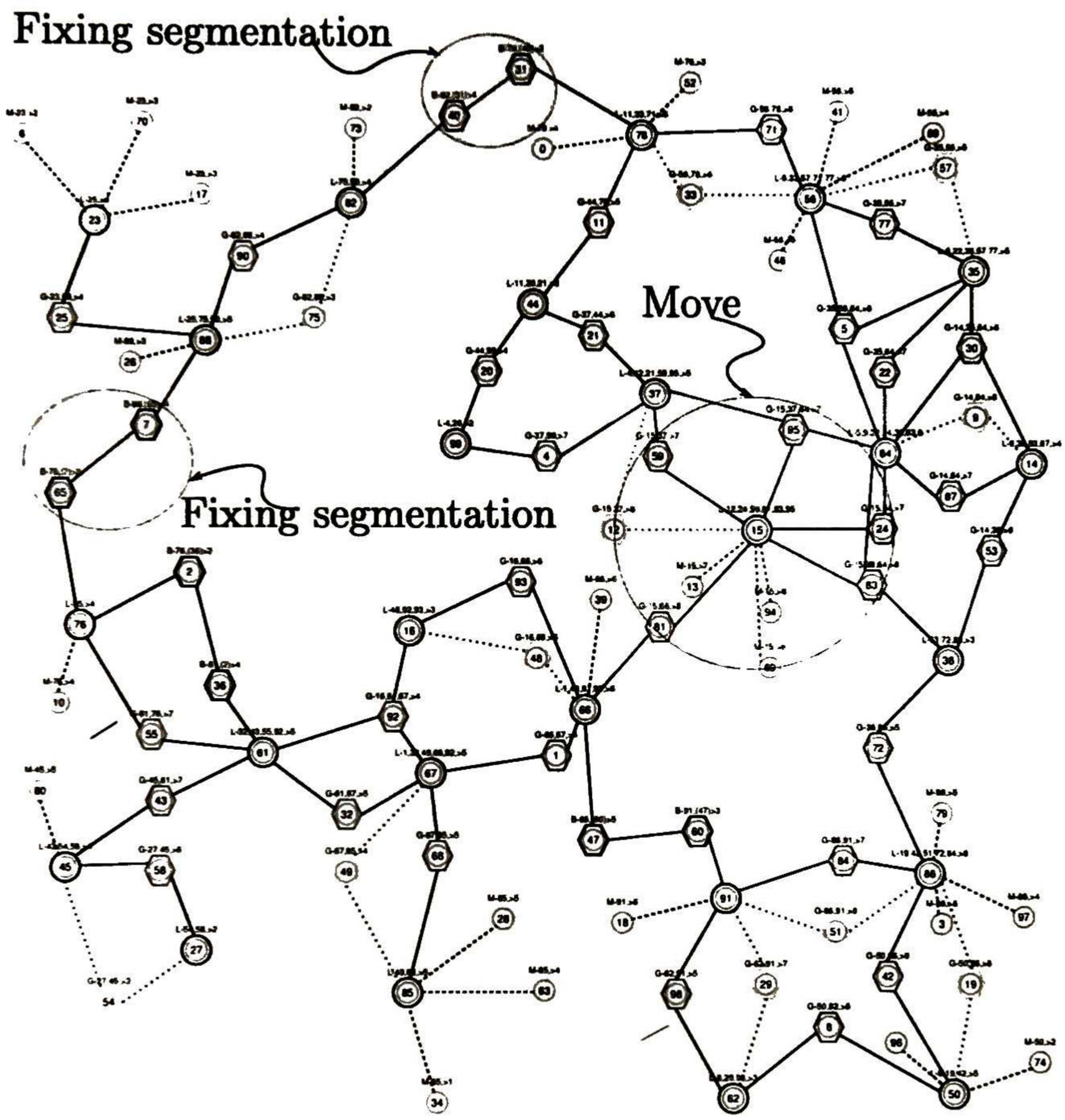

Figure 4.11: Initial dynamic backbone - Snapshot before the movement of leader agents

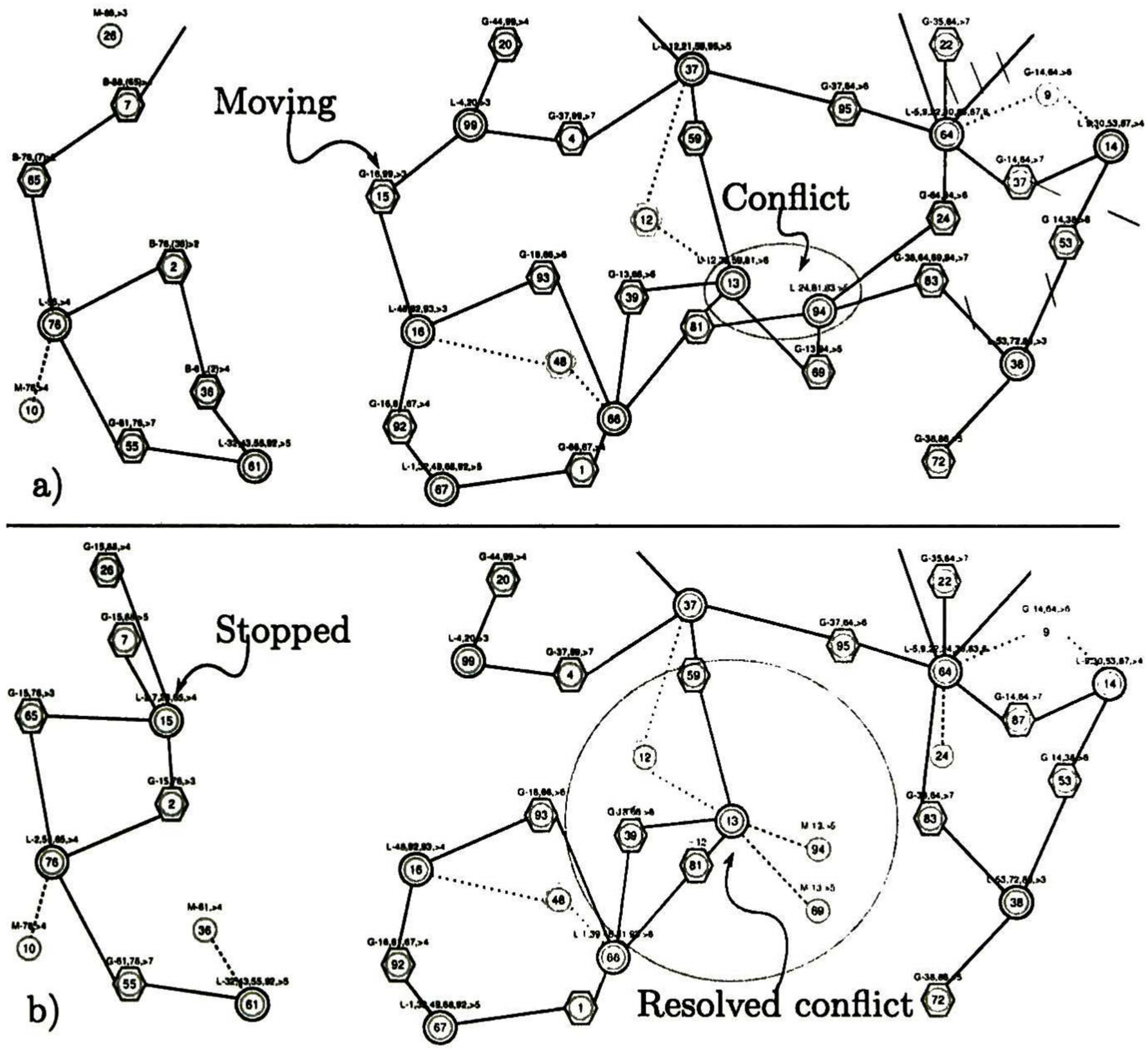

Figure 4.12: A leader movement - Movement of the leader agent with id 15

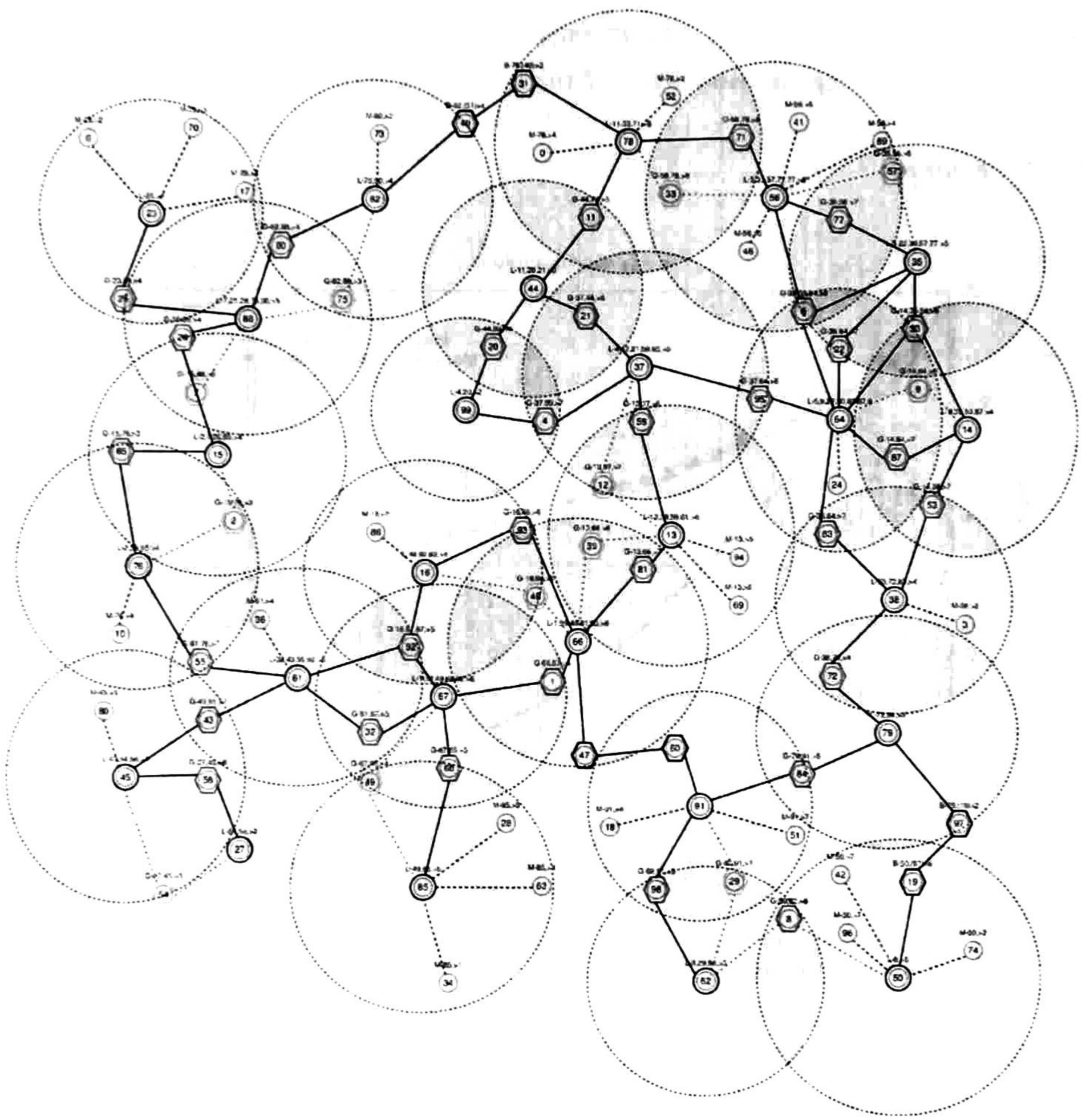

Figure 4.13: Structure after movement - Snapshot after the movement of some leader agents

gets the leadership is the agent 3, but when the group is reorganizing the agent 3 move towards the agent $38's$ group.

Figure 4.13 shows a snapshot after the movement of agents. In this example the robustness to agent mobility of the proposed strategy is demonstrated. The exhibited behavior shows the adaptive capabilities of the involved agents.

4.2.4.3 Energy

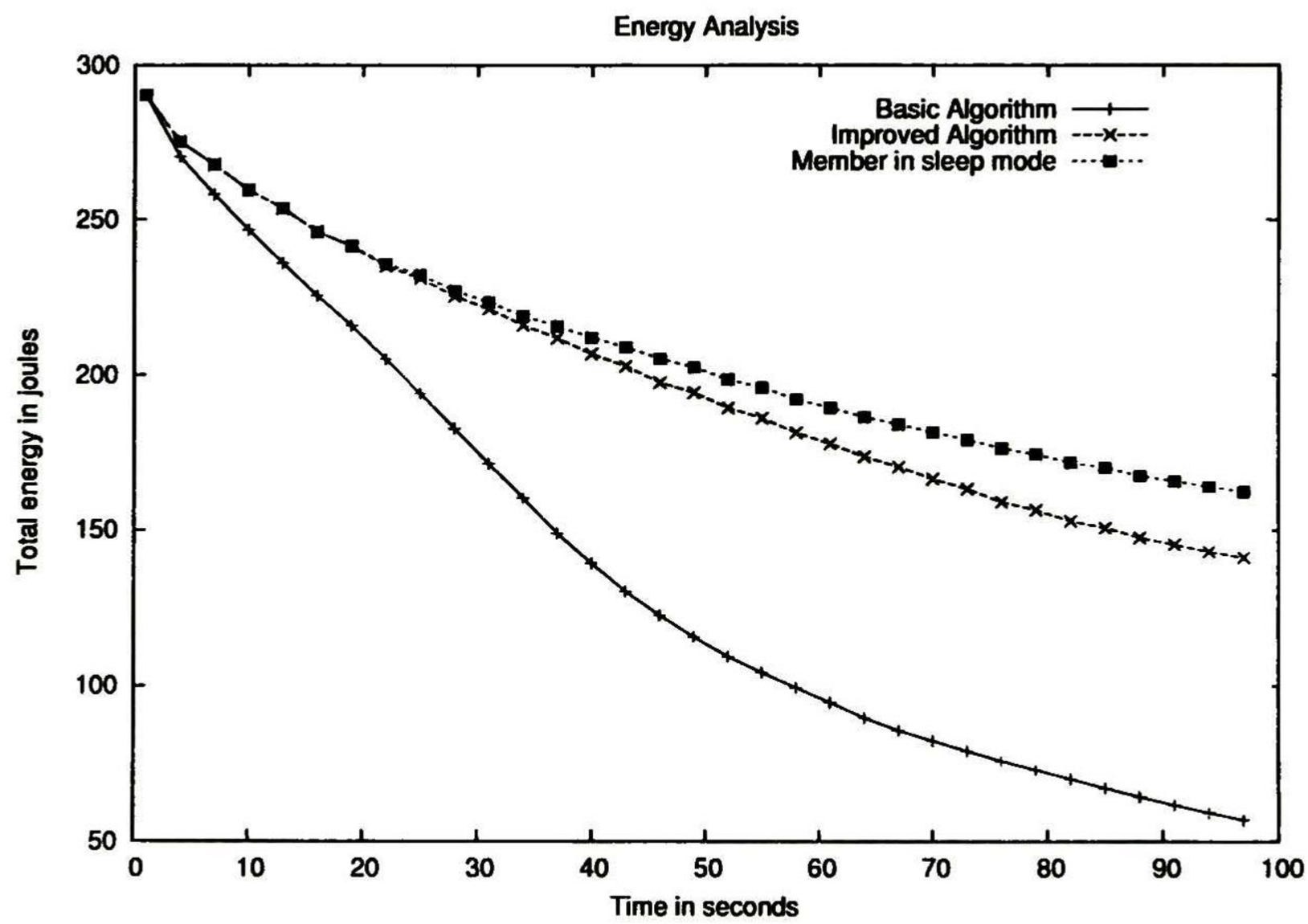

Figure 4.14: Remaining Energy - Remaining energy of diverse strategies

The total energy usage is defined as the total sum of each individual energy usage of each agent in the interval $[0, \tau]$, where τ is the maximum simulation time.

Figure 4.14 shows the performance comparison of different versions of the algorithm. It is obvious the decreasing trend of the three curves; this implies that the consumption of energy gets bigger when we do not make any enhancement. The lower curve in the graph shows the energy consumption using the basic algorithm. This algorithm only forms the virtual backbone regardless the remaining energy. The next upper curve shows the energy performance of the proposed algorithm.

As we stated earlier, according to the environment sometimes is not necessary to maintain all the member agents in active mode to ensure the communication. The last upper curve shows important energy saving, if member agents are turned off after they get their role.

4.2.4.4 Roles

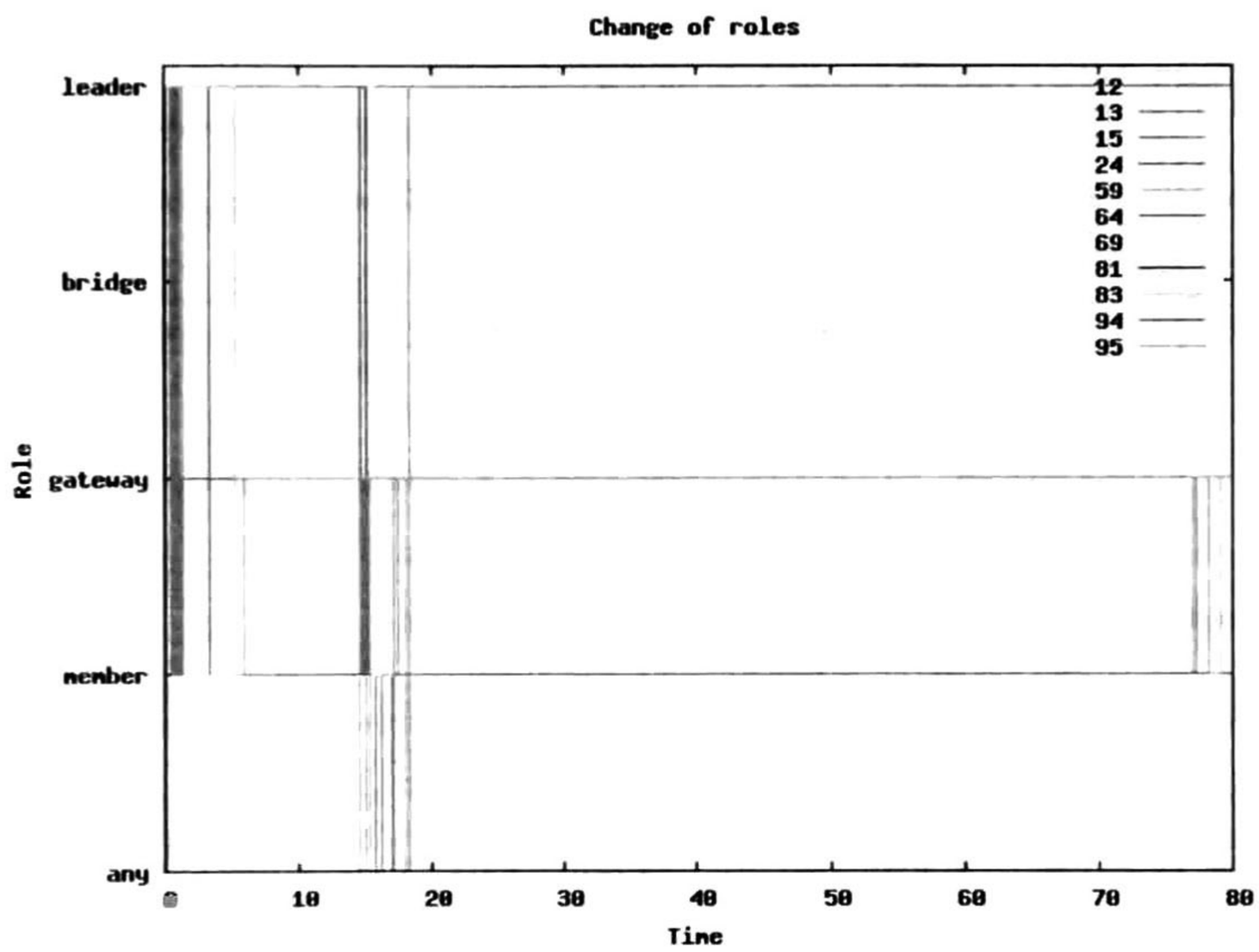

Figure 4.15: Roles - Agent's role switching during operation

Figure 4.15 illustrates the behavior based on the role changes of the group that is represented by the leader agent with id 15. Note that at the beginning of simulation, the agents take their role, and then they are stabilized. After that, in the interval of time between $t = 14$ and $t = 20$, the three agents mentioned earlier are moved. As you can notice, there are some changes of role. Even at times the agent 15 take the "Any" role. After environment changes the agents are stabilized. By $t = 80$ some agents change their role because of some agents died.

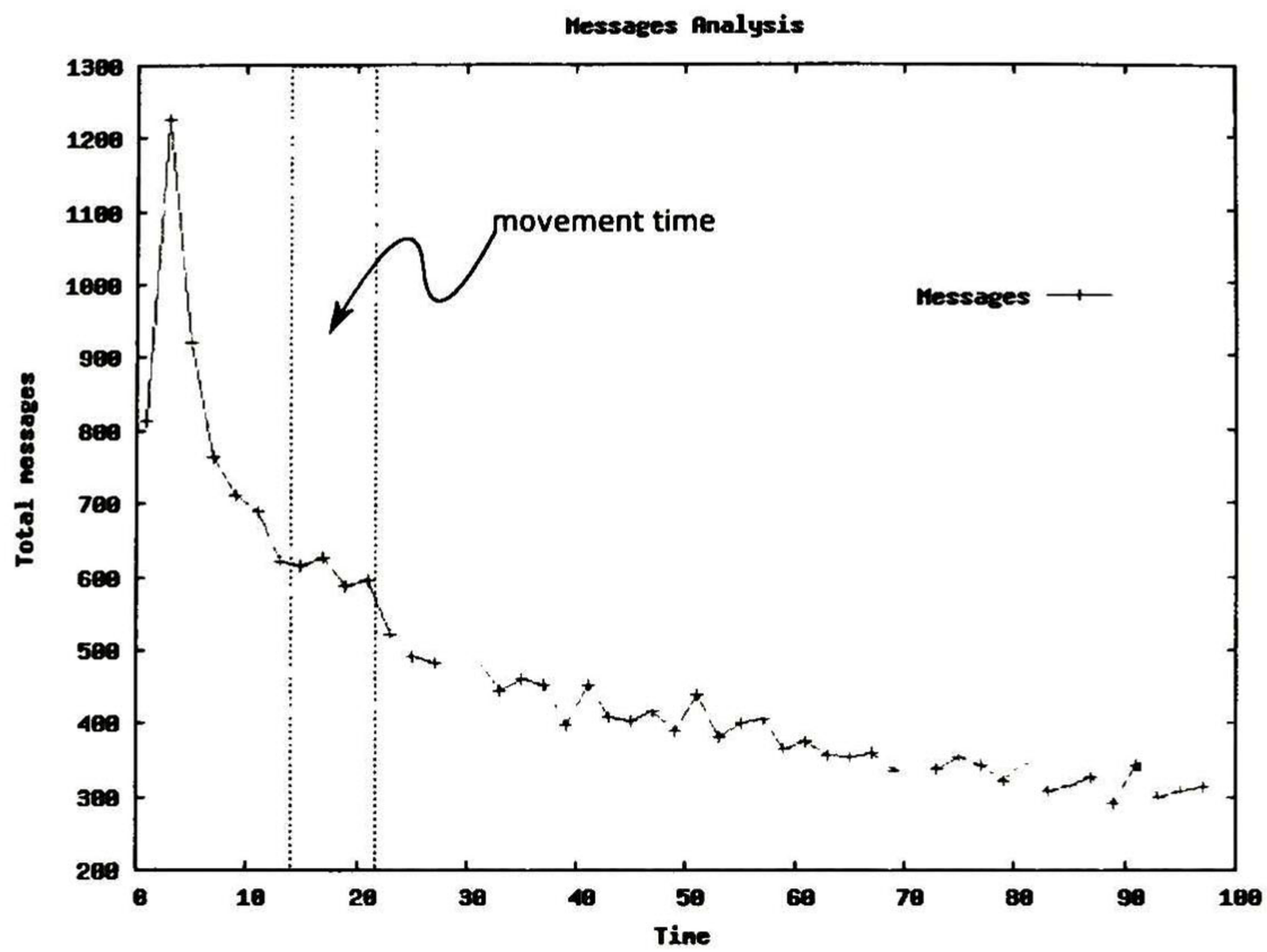

Figure 4.16: Messages - Messages in the entire network

4.2.4.5 Messages

At the beginning each agent has to discover its neighborhood. Each agent maintains a list of all its neighbors. An agent sends a *hello* message asking for neighbors. Then, when an agent receives this message, it responds with an *acknowledge* message. In order to ensure a bidirectional communication, the agents that receive an *acknowledge* message make a link between them. Due to this fact the number of messages is higher at start up than the exchange of messages when the network is stabilized. This can be seen in Figure 4.16, where the number of messages is decreasing as time passes until the neighbors table does not change.

4.3 EESOA Simulations

In this section, we conduct several simulations to assess the performance of EESOA. We evaluate the pure protocol structure brought by the strategy: therefore no data traffic is generated in the network.

The simulation parameters are described as follows:

- The agents are placed within an area of 100 × 100 meters.
- Each agent is randomly deployed.
- The radio interface used for simulations is the standard 802.15.4.
- Transmission range: 15 meters.
- Each simulation is executed for 100 seconds.

The algorithm EESOA is compared against two different distributed algorithms namely Multi-Wireless-Agent Communication (MWAC) (26) and Guaranteed Routing Cost Minimum Connected Dominating Set (GOC-MCDS) (16).

MWAC is chosen because the structure MWAC builds is similar to that EESOA builds. MWAC exploits the emergence resulting from a multi-agent self-organization process, where a group is formed by one group representative agent (leader), some connection agents (gateways) and possibly some simple members.

Hongwei et. al. in (16) present GOC-MCDS; it is a polynomial time algorithm which produces a CDS D of size at most $\alpha \cdot opt_{MCDS}$ for some fixed constant α and with property that for every pair of nodes u and v, there is a path between u and v with intermediate nodes in D and length at most $5 \cdot d(u, v)$. EESOA is compared with the distributed version of GOC-MCDS, which has the same performance ratio as the centralized algorithm. The distributed version comprises of two stages: one is construct a maximal independent set as dominating set; another is to connect the dominating set by choosing intermediate nodes.

In order to do a fair comparison, the three algorithms start at the same time after neighborhood was discovered. Then, they start at instant τ.

4.3.1 Simulation Results

In order to show the different virtual structures generated by the three algorithms, an example is showed in Figures 4.17-4.19. After running each algorithm, Figures 4.17,4.18 and 4.19 show the virtual structure built by EESOA, MWAC and GOC-MCDS algorithm respectively. The backbone built by EESOA communicates the complete network using 3 couple of bridges and with one leader less than MWAC (see Figure 4.17), instead, the final backbone formed by MWAC can communicate the whole network with

22 leaders (see Figure 4.18). GOC-MCDS has much more leaders (black nodes) than EESOA and MWAC because it has the condition of shortest path between a node pair in the network. In addition, we can observe that not all virtual backbones have bridge agents, and on average the number of bridges per simulation is approximately 6 in an environment with 100 agent nodes. Finally, Figures 4.17,4.18 and 4.19 also show the difference on the number of gateways and members in EESOA, MWAC and GOC-MCDS respectively.

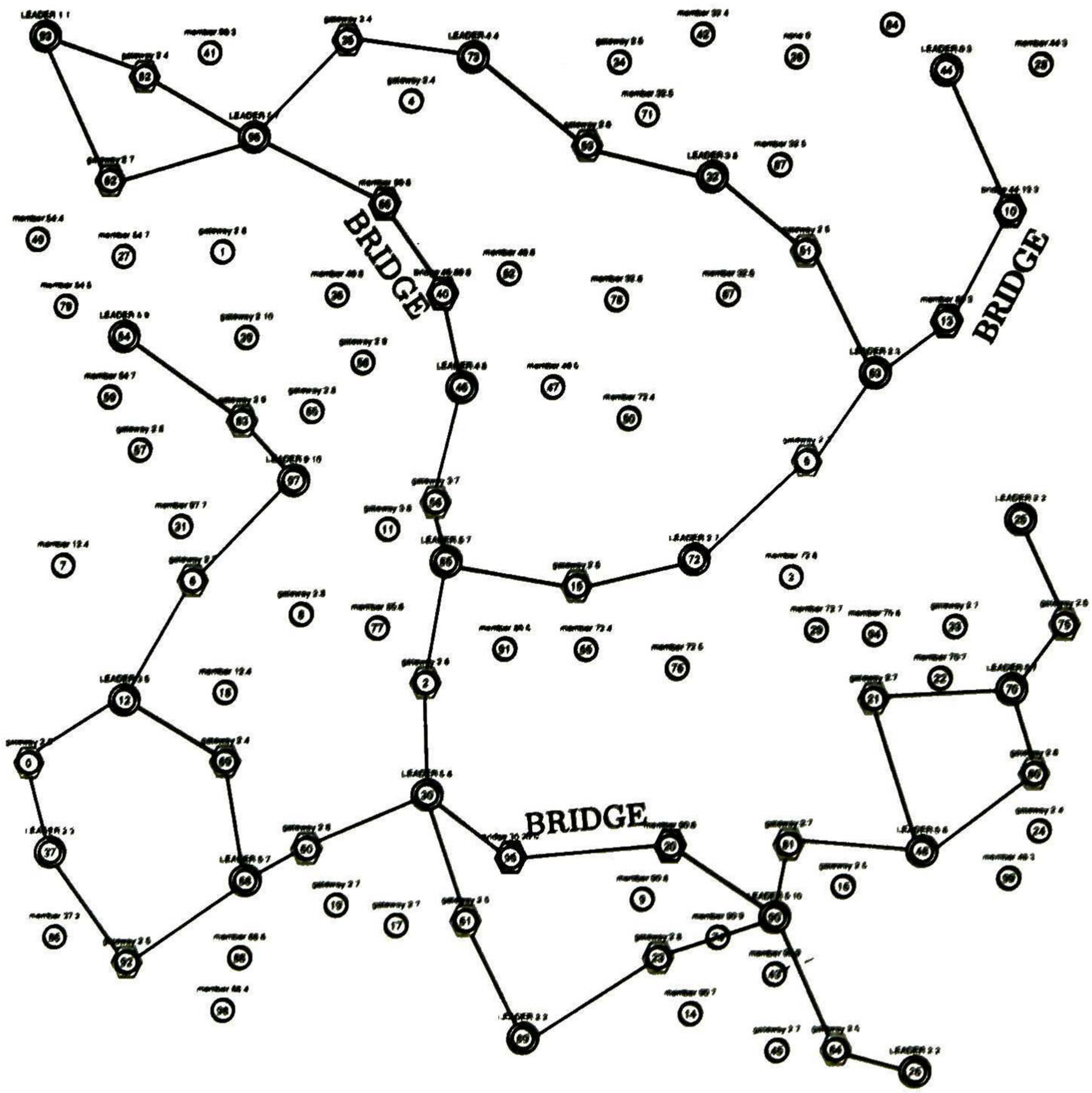

Figure 4.17: EESOA - It has 21 leaders, 22 gateways, 51 members and 3 pairs of bridge agents.

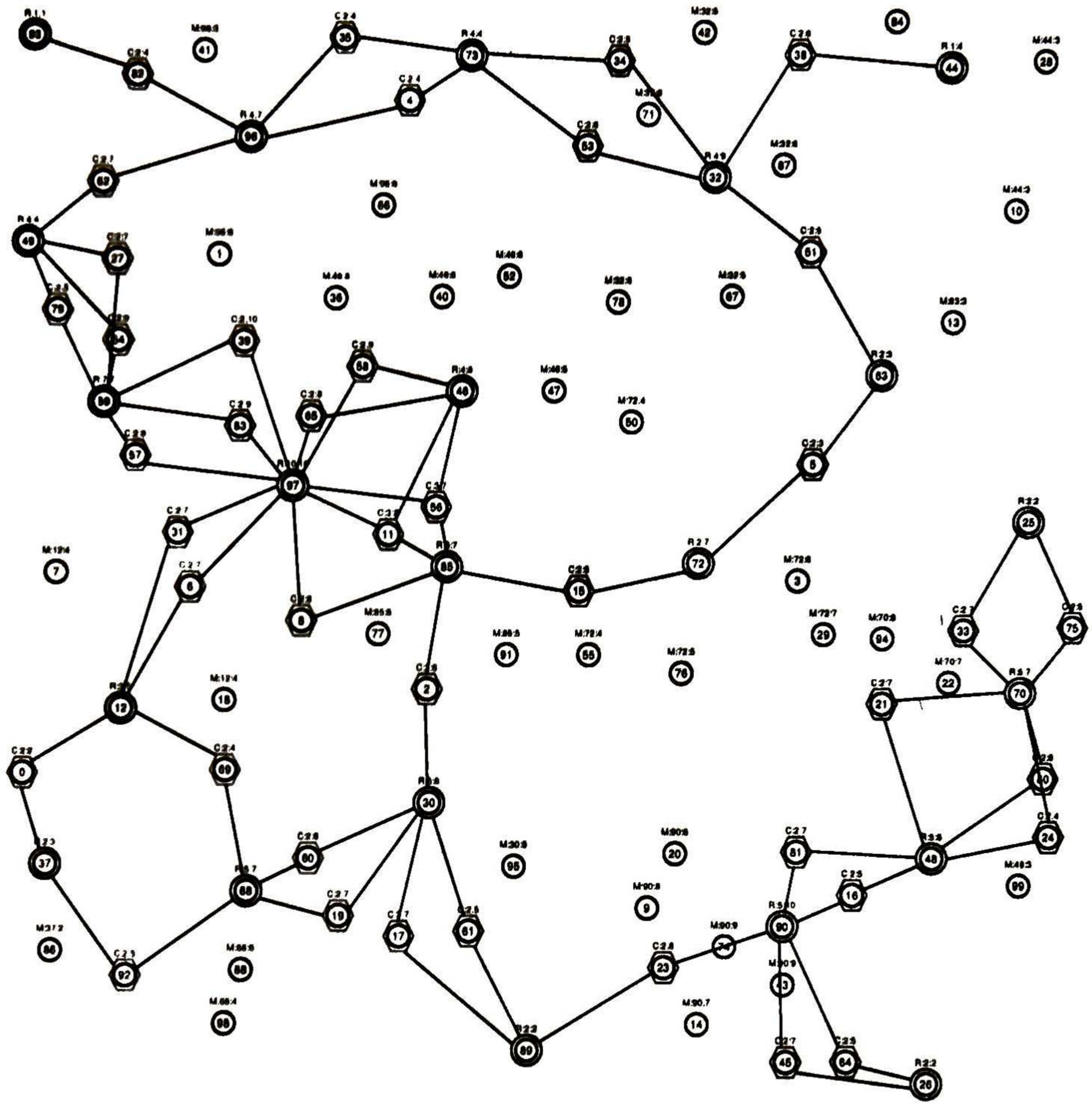

Figure 4.18: MWAC - It has 22 leaders, 41 gateways and 37 members.

4.3.2 Virtual Backbone Properties

For our experiments 20 runs were made. In addition, each run represents the random deployment of the 100 agent nodes in the simulation area and the building of the virtual backbone. The simulation parameters defined for EESOA algorithm are described in table 1.2.

From this initial set up, it can be noticed in Figure 4.20 that EESOA has on

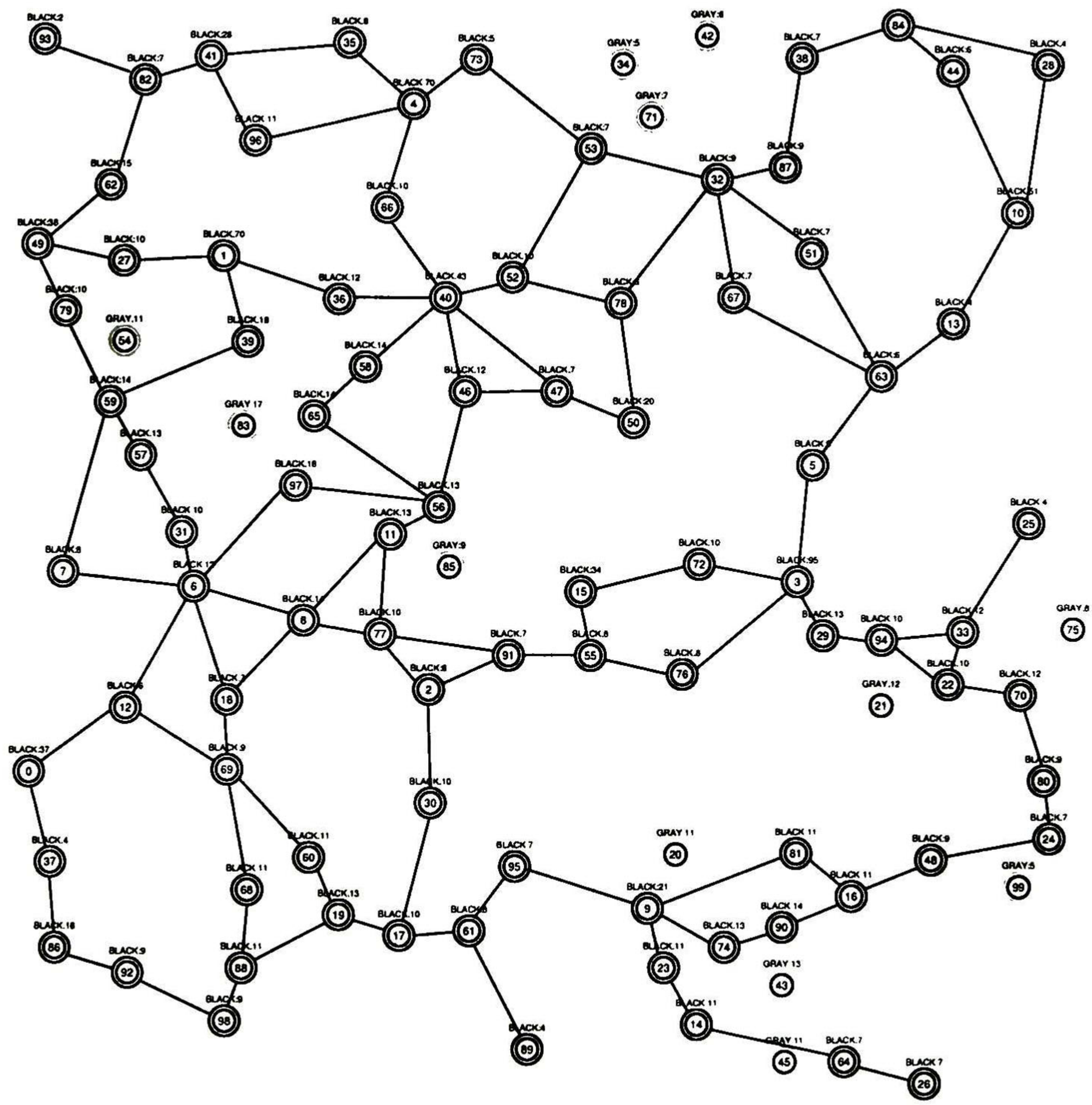

Figure 4.19: GOC-MCDS - It has 29 leaders (black nodes), 59 gateways and 12 members.

Table 4.2: Algorithm parameters

Symbol	Value in seconds	Remarks
ξ	1.5	a value to schedule the broadcast
T	1	broadcast interval
ρ	3	frequency count
κ	5	elapse time as an agent keeps a role
Δ	2	a constant value
τ	10	starting time of algorithms

average a lower number of leaders per simulation run. MWAC was better only in 3 of 20 simulation runs than EESOA, the same behavior can be shown with GOC-MCDS. This can be explained by the fact that EESOA chooses the more suitable agents to be leaders using the inhibition concept, which means leaders cover the greatest number of members in the neighborhood. On the contrary, GOC-MCDS chooses the nodes regarding the lowest *id* of nodes in the neighborhood. Even though, MWAC follows the priority strategy for choosing representatives, if an agent takes the representative role, no one else can be a representative in the neighborhood. Thus, the representative agent could not be the best among the neighbors. MWAC also exhibits a segmentation problem. Without making any reorganization in order to solve the segmentation. Figure 4.24 also shows the number of segments MWAC has per simulation.

In order to save energy as much as possible, the strategy of choosing only one gateway to make the communication among the same leaders is carried out. Figure 4.21 shows the number of gateways that MWAC has per simulation. It can be seen that MWAC has on average 40 gateways against 22 gateways in EESOA. The number of gateways slightly increases in GOC-MCDS because the algorithm tries to obtain the minimum routing cost. In general this means that MWAC and GOC-MCDS will spend on average more energy than EESOA.

Member agents are responsible for their own tasks, and they do not retransmit packets. This is why it is desirable to have as many member agents as possible per group. The number of member agents per simulation is shown in Figure 4.22. It can be seen that the number of members in EESOA is greater than the one in MWAC and

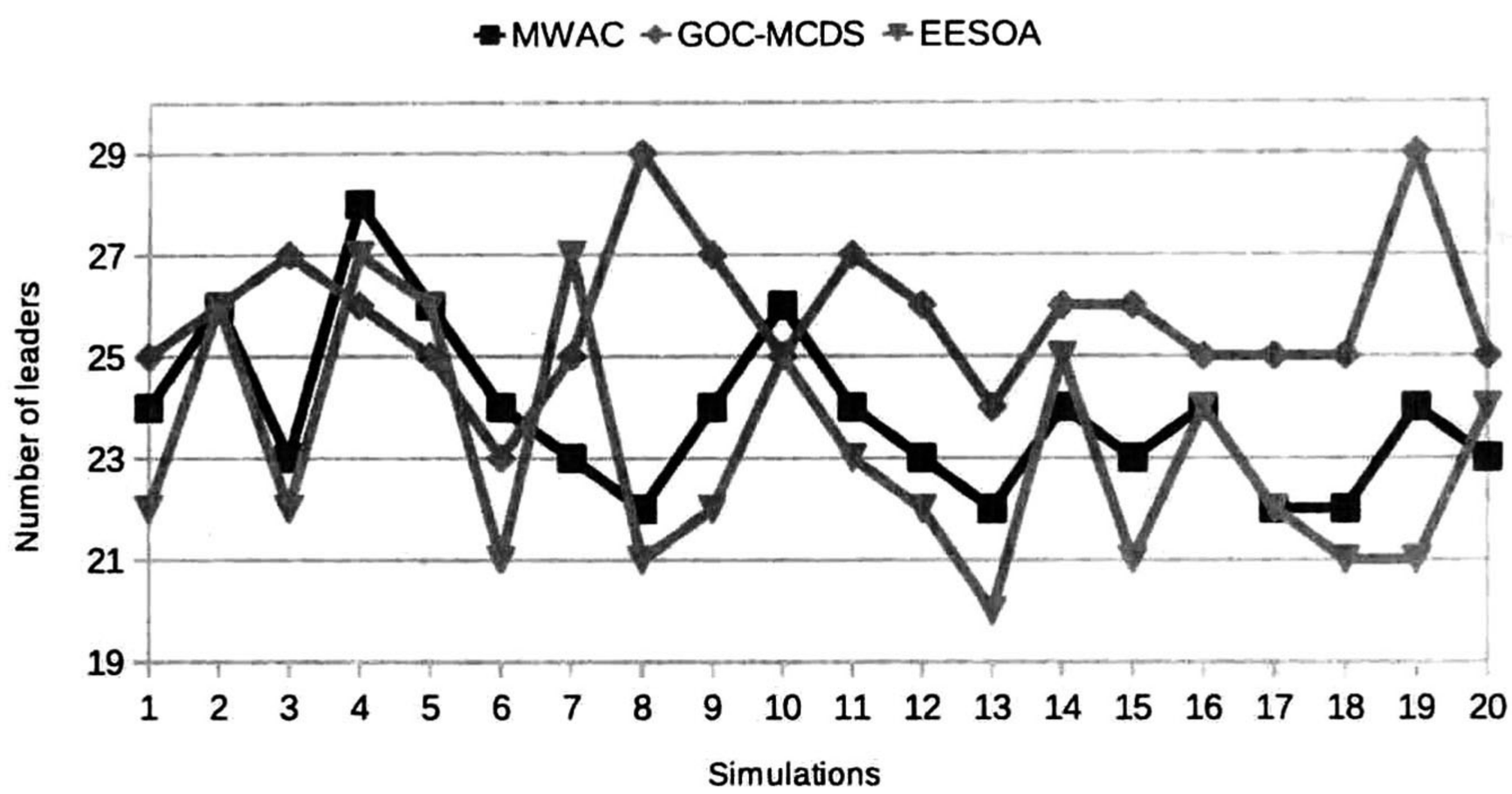

Figure 4.20: Leaders - Number of leaders per simulation.

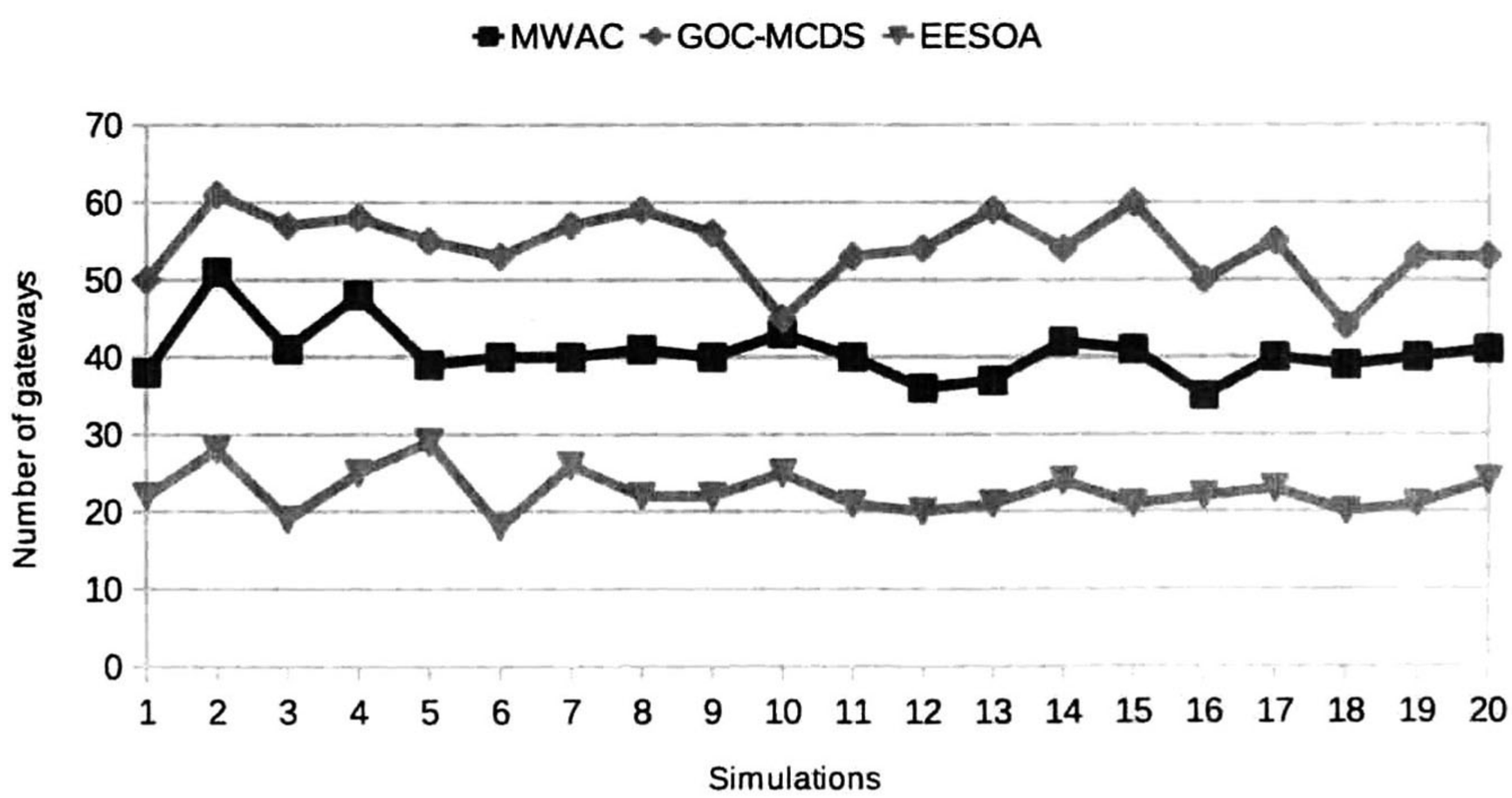

Figure 4.21: Gateways - Number of gateways per simulation.

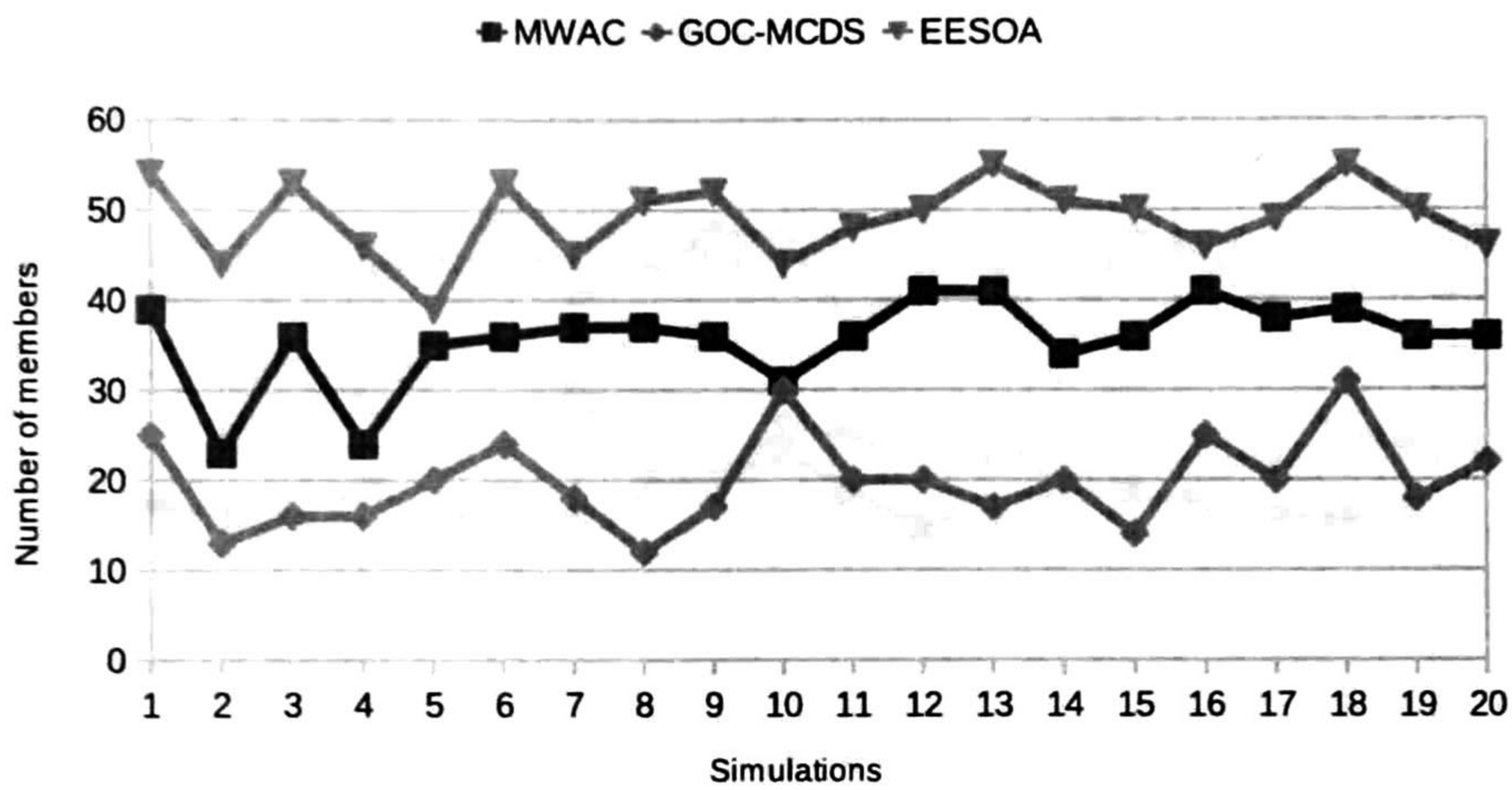

Figure 4.22: Members - Number of members per simulation.

GOC-MCDS. EESOA has almost an average of 50 member agents against an average of 35 and 20 member agents in MWAC and GOC-MCS per simulation run respectively. All this shows how EESOA outperforms MWAC and GOC-MCDS in the building of the virtual backbone.

Figure 4.23 shows the remaining energy per simulation of the MWAC and GOC-MCDS algorithm and the proposed algorithm respectively. The energy comparison was only done with the energy consumption due to the construction and maintenance of the network. MWAC has a remaining energy average of 26%, while the proposed strategy has a remaining energy average of 55%. This means that EESOA can obtain an energy saving of 29% regarding to MWAC. It can be seen that GOC-MCDS exhibits a better result than MWAC based on energy consumption, this is because GOC-MCDS does not broadcast extra *hello* messages after the steps of the algorithm have finished. This is a drawback because GOC-MCDS does not take into consideration the reconfiguration when a node joins the network.

Figure 4.24 provides the comparison of MWAC and GOC-MCDS in terms of number of bridges and segments in the virtual structure. Since MWAC does not use bridge agents, the virtual structure can result in segments. Because the EESOA algorithm is

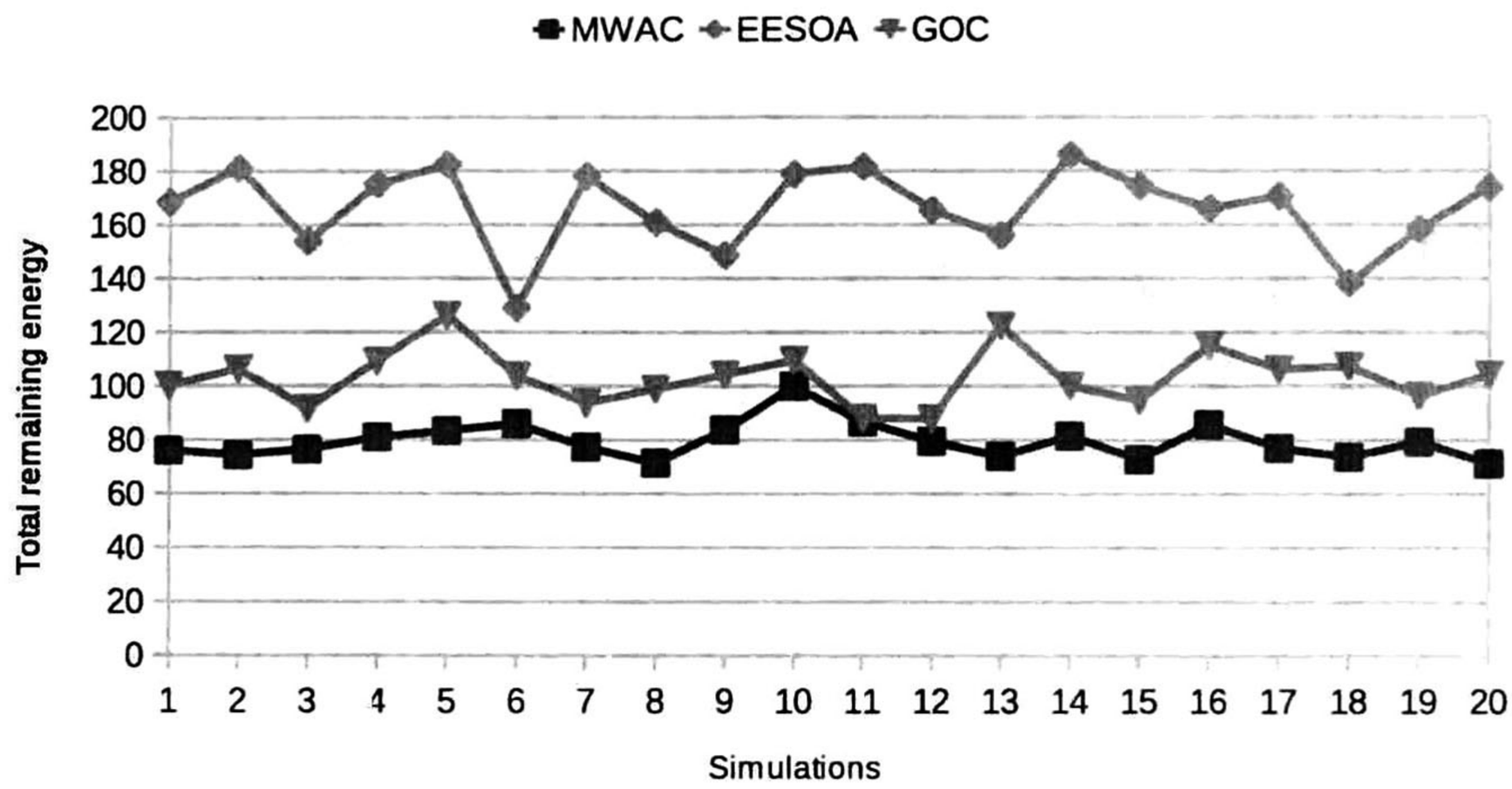

Figure 4.23: Remaining energy - Remaining energy of the different strategies.

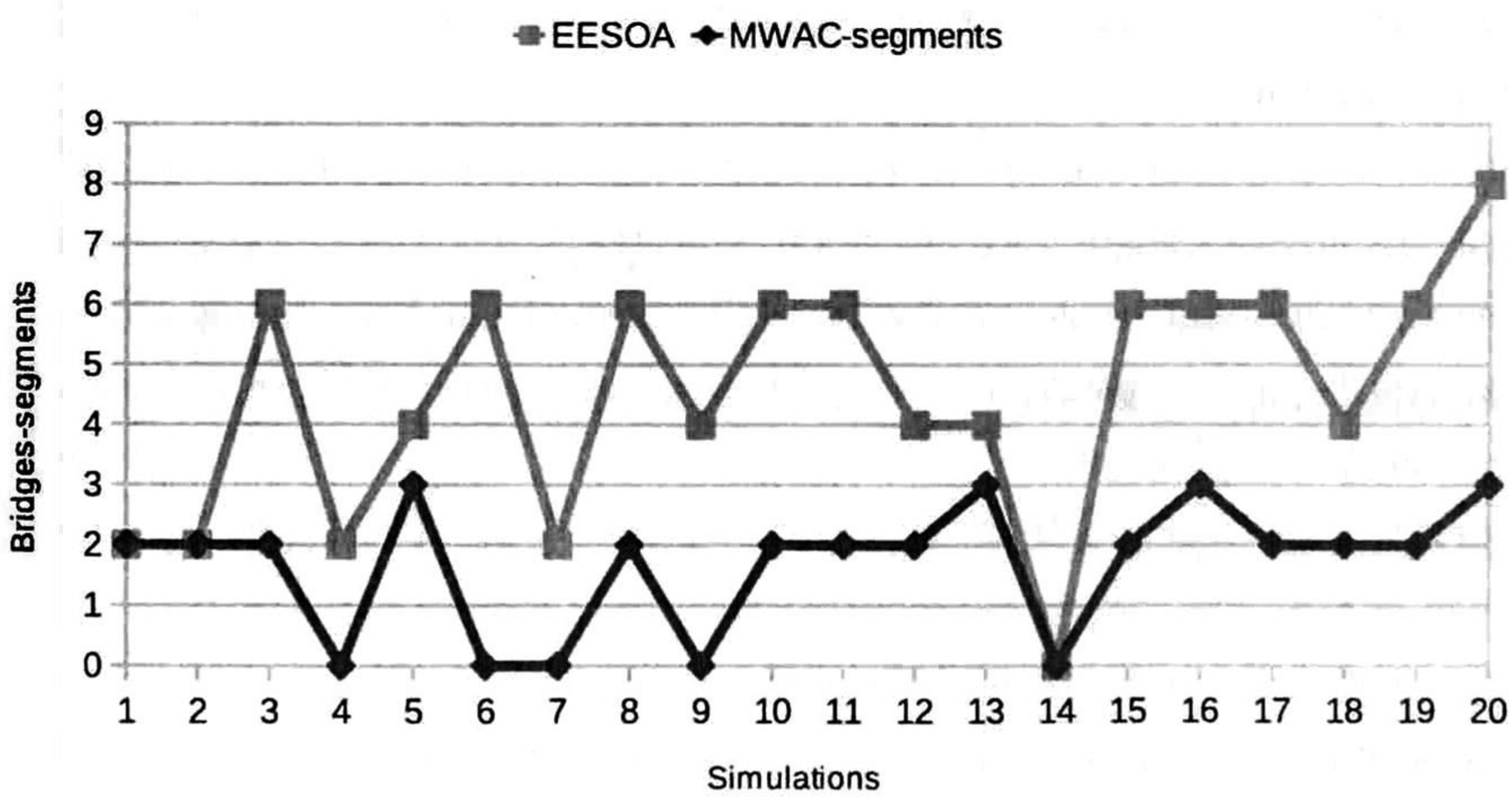

Figure 4.24: Bridges and segments - Number of bridges and segments in EESOA and MWAC respectively.

localized, as well as MWAC, most of the time some member agents switch to bridge in order to communicate possible segments of the network. On average there exist three pair of bridge agents in an environment of 100 agents.

4.4 Comparing Global Against Distributed Approaches

4.4.1 Minimum Spanning Tree

As a first approach, MWAC-based strategy is compared with a defined MST, which is weighted appropriately to obtain the capacity of agents (see section 2.1.1).

Minimizing links cardinality in the computed virtual backbone can help to decrease the control overhead since broadcasting for route discovery and topology update is restricted to a small subset of agents. Therefore, the broadcasting storm problem inherent to global flooding can be greatly decreased. In this way, virtual backbone provides good resource conservation property.

In order to evaluate the network formed by the MWAC-based strategy, we compare it with the one obtained using the MST global procedure. Then, the number of links is compared.

Results for several scenarios are summarized in table 4.3. Additionally, the number of links forming the backbone is compared. It is noticeable that the number of links is close to the minimal required for obtaining connectivity in these simulations. In each simulation that has been done, it is possible to show that every agent in the MST_B^T set is in the virtual backbone generated by the proposed strategy. Hence, it is evident the high resemblance between the MST_B^T structure and the one generated by the self-organized strategy.

4.4.2 Particle Swarm Optimization

Because the structure formed by the MST is not the same that the one formed by the distributed algorithms, it is necessary another global approach, i.e., it is necessary a structure built by nodes that have the role of leader and gateway. Then, in order to obtain a better approximation to the structure formed by the distributed algorithm, another global approach was designed.

As described in chapter 3, PSO is a global method that is used for building an optimal backbone. On the contrary, EESOA is a distributed algorithm that uses only local

Table 4.3: MST comparative table

Environment		No. links			Performance		
Agents	Dim.	MST	MWAC-based	UDG	Inactive L.	#Links diff.	Proximity
50	50x50	49	56	97	3	4	92%
60	100x100	59	68	101	3	6	90%
70	100x100	69	88	115	2	17	80%
80	100x100	79	96	146	9	8	90%
100	100x100	99	125	256	10	16	86%

information, which is described in section 2.4. This section provides the comparison between two algorithms, the global and distributed approach respectively.

Figure 4.25 shows the comparison based in number of leaders and segments per simulation. It can be seen that in some of the simulations the number of leaders between PSO and EESOA are equal. It is worth mentioning that could there be other configurations with lower number of leaders, but PSO search for a solution that have a trade-off between the different objective functions.

The number of gateways per simulation is shown in Figure 4.26. The result of PSO and EESOA algorithms is quite similar. This similarity comes because EESOA uses a method to deal with redundant gateways.

Figure 4.27 shows the comparison based on number of members per simulation. There, every redundant gateway becomes member. Therefore, the number of members per simulation is on average the same that the result of PSO.

4.5 Conclusion

Although, this is the first approximation of the algorithm, the simulation results show that it outperforms the self-organized algorithm MWAC and the distributed GOC-MCDS algorithm. Simulations also show that the distributed algorithm EESOA can obtain energy-efficient well-formed structures compared with the structures built by a global solution.

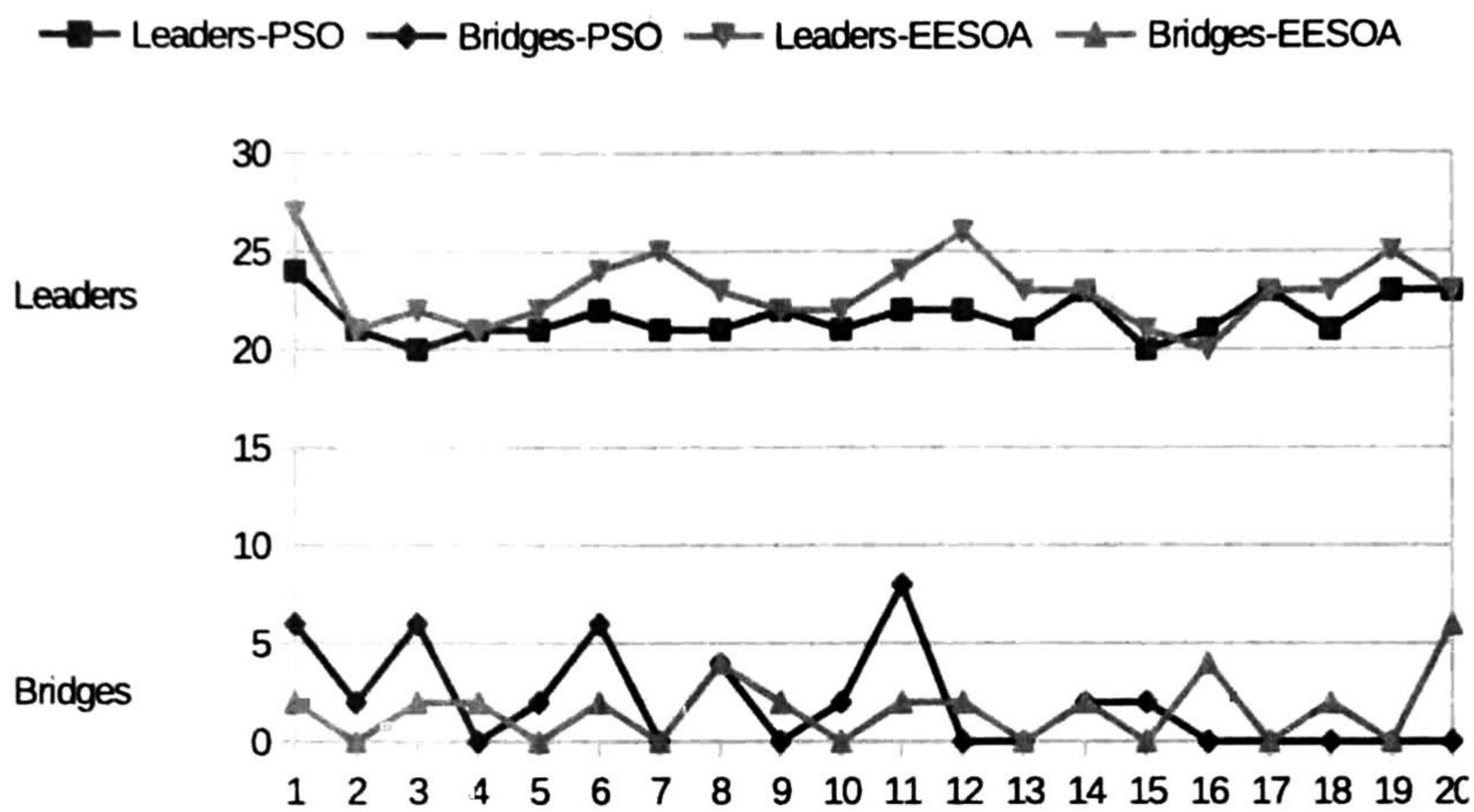

Figure 4.25: Leaders PSO-EESOA - Number of leaders and segments per simulation.

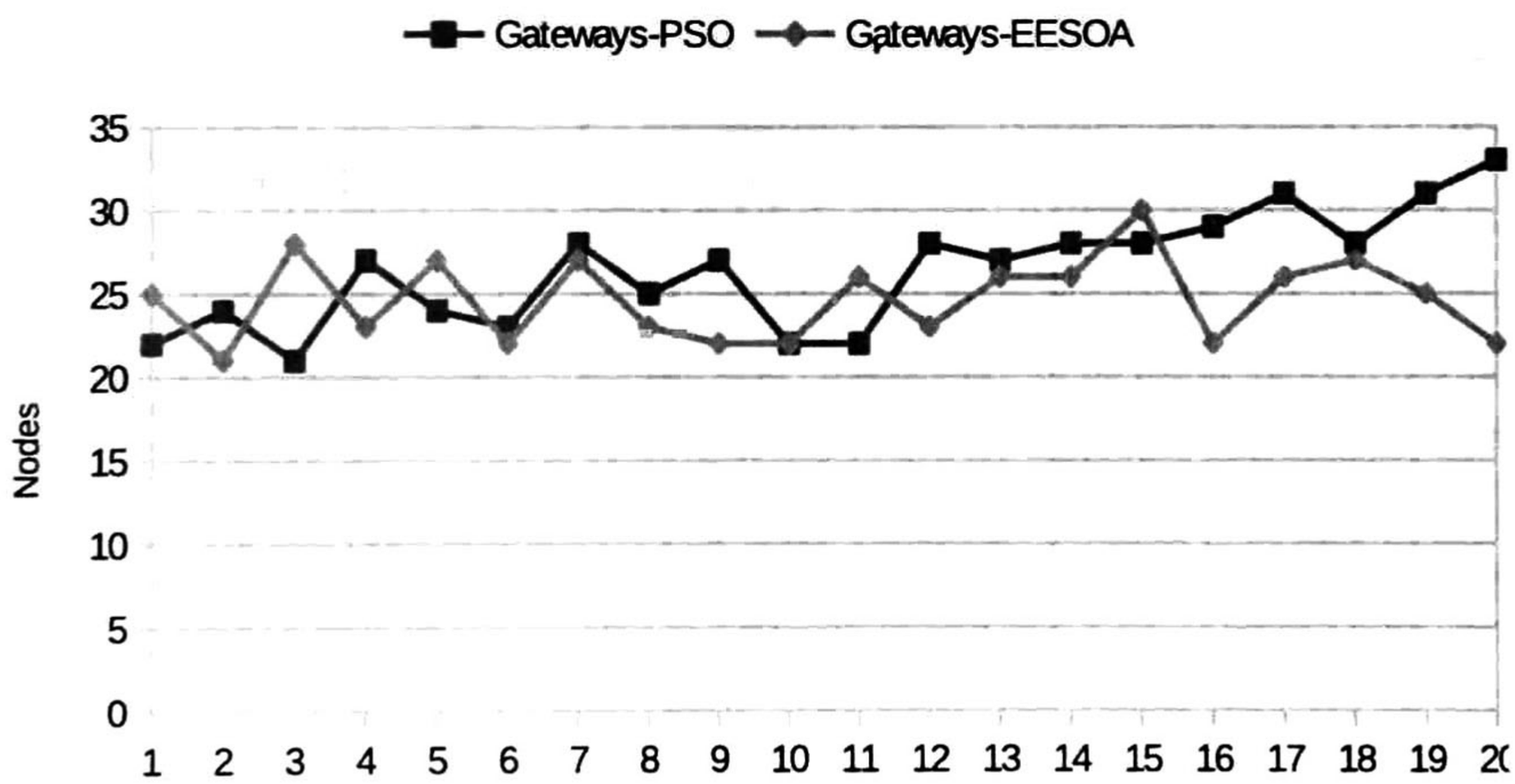

Figure 4.26: Gateways PSO-EESOA - Number of gateways per simulation.

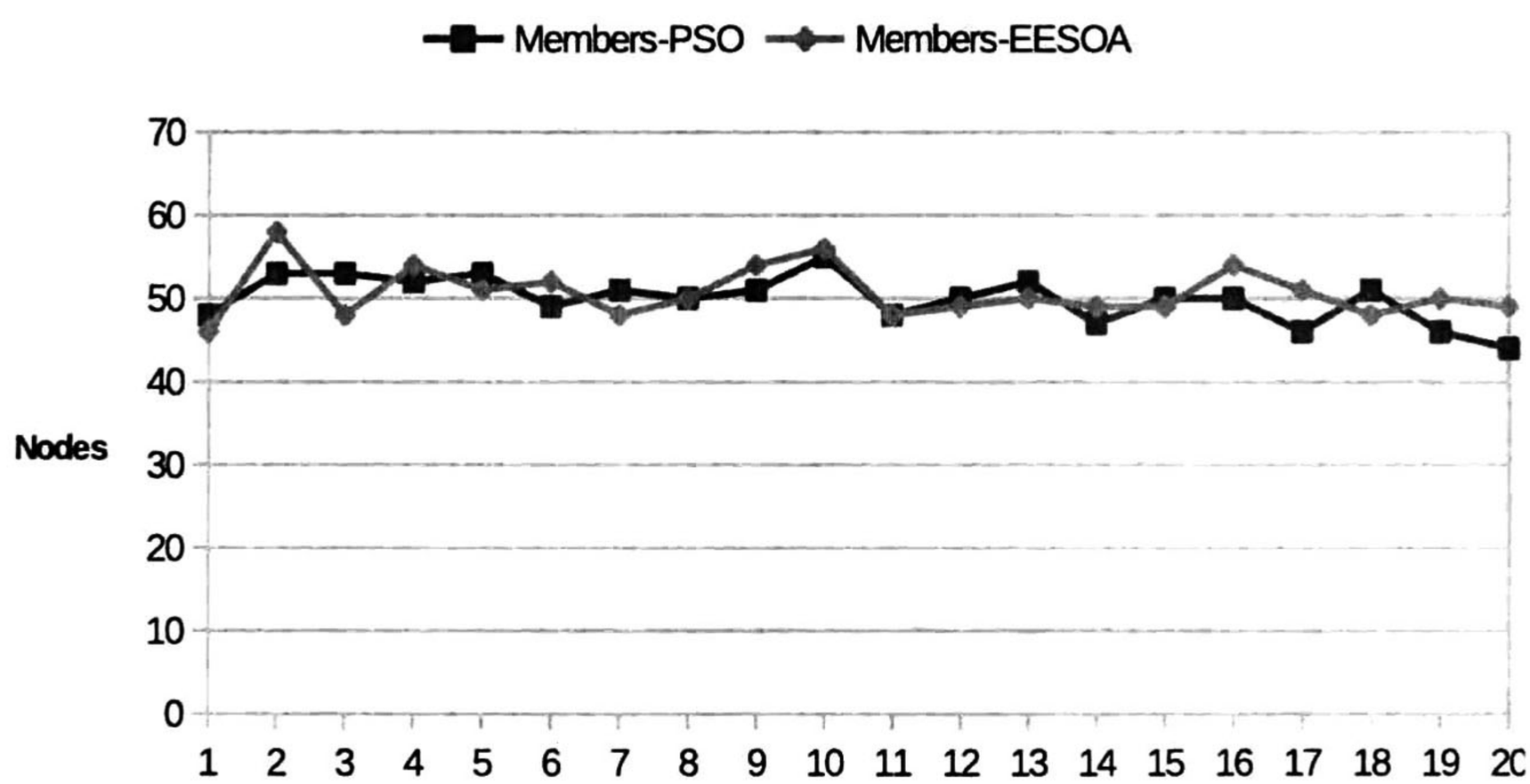

Figure 4.27: Members PSO-EESOA - Number of members per simulation.

Chapter 5

Concluding Remarks

Multi-agent based approach using self-organization strategies allows network formation with efficient maintenance. Important savings in energy consumption has been obtained through the proposed group-based distributed algorithm, which handles variations on both transmission power and transmission periods. Furthermore, role switching during the agents operation increases the network lifetime, coping with dynamic behavior of the nodes population such as node mobility, node losses, and new node arrival. Simulations show substantial improvements with respect the basic algorithm, which do not use dynamic power transmission.

This work also states the problem as a multi-objective optimization on a graph; then it uses a particle swarm optimization algorithm to obtain an optimal solution. Furthermore, we have created a method for calculating an optimal virtual backbone among the nodes taking advantage of the independence property. The multi-objective approach allows representing the different functional constraints under the problem of virtual backbone generation. The simulations show that this approach is efficient and effective to obtain a well spread solutions in the Pareto frontier. However, further research needs to be done to compare this algorithm with the more traditional methods.

5. CONCLUDING REMARKS

References

[1] Fuad Bajaber and Irfan Awan. Adaptive decentralized re-clustering protocol for wireless sensor networks. *Journal of Computer and System Sciences*, 2010. ix

[2] Stefano Basagni, Imrich Chlamtac, Andras. Farago, and Erik Jonsson. A generalized clustering algorithm for peer-to-peer networks. In *in Workshop on Algorithmic Aspects of Communication*, 1997. 33

[3] Jeremy Blum, Min Ding, Andrew Thaeler, and Xiuzhen Cheng. Connected dominating set in sensor networks and manets. *Handbook of Combinatorial Optimization*, pages 329–369, 2004. 13, 33

[4] Alberto Cerpa, Jennifer L. Wong, Louane Kuang, Miodrag Potkonjak, and Deborah Estrin. Statistical model of lossy links in wireless sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press. ISBN 0-7803-9202-7. URL `http://dl.acm.org/citation.cfm?id=1147685.1147701`. 28

[5] J.Jasper Gnana Chandran and S. P. Victor. Optimized energy efficient localization technique in mobile sensor networks. *IACSIT International Journal of Engineering and Technology*, 2(2): 149–156, 2010. ISSN 1793-8236. 34

[6] J.Jasper Gnana Chandran and S. P. Victor. An energy efficient localization technique using particle swarm optimization in mobile wireless sensor networks. *American Journal of Scientific Research*, 2010. 34

[7] Carlos A. Coello Coello, Clarisse Dhaenens, and Laetitia Jourdan, editors. *Advances in Multi-Objective Nature Inspired Computing*, volume 272 of *Studies in Computational Intelligence*. Springer, 2010. ISBN 978-3-642-11217-1. 39, 48

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001. 12

[9] Luiz H.A. Correia, Daniel F. Macedo, Aldri L. dos Santos, Antonio A.F. Loureiro, and José Marcos S. Nogueira. Transmission power control techniques for wireless sensor networks. *Computer Networks*, 51(17):4765–4779, July 2007. 19

[10] Douglas S. J. De Couto, Daniel Aguayo. Benjamin A. Chambers, and Robert Morris. Performance of multihop wireless networks: shortest path is not enough. *Computer Communication Review*, 33(1):83–88, 2003. 29

[11] DARPA. URL `http://www.isi.edu/nsnam/ns/`. 53

[12] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001. 39, 42

[13] Mieso K. Denko. The use of mobile agents for clustering in mobile ad hoc networks. In *Proceedings of the annual research conference of SAICSIT*, pages 241–247, Republic of South Africa, 2003. ISBN 1-58113-774-5. 7

[14] Nikos Dimokas, Dimitrios Katsaros, and Yannis Manolopoulos. Node clustering in wireless sensor networks by considering structural characteristics of the network graph. In *ITNG: Proceedings of the International Conference on Information Technology*, pages 122–127, Washington, DC, USA, 2007. ISBN 0-7695-2776-0. ix

[15] Falko Dressler. A study of self-organization mechanisms in ad hoc and sensor networks. *Computer Communications*, 31(13):3018–3029, Feb 2008. Special Issue:Self-organization and self-management in communications as applied to autonomic networks. ix

[16] Hongwei Du, Qiang Ye, Weili Wu, Wonjun Lee, Deying Li, Dingzhu Du, and S. Howard. Constant approximation for virtual backbone con-

struction with guaranteed routing cost in wireless sensor networks. In *INFOCOM, Proceedings IEEE*, pages 1737–1744, april 2011. doi: 10.1109/INFCOM.2011.5934967. 73

[17] A. Farhang-Mehr and S. Azarm. Diversity assessment of pareto optimal solution sets: an entropy approach. In *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02*, volume 1, pages 723–728, May 2002. doi: 10.1109/CEC.2002.1007015. 48, 52

[18] M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. 1979. ix, 3, 6

[19] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990. ISBN 0716710455. 36

[20] Jing He, Zhipeng Cai, Shouling Ji, Raheem A. Beyah, and Yi Pan. A genetic algorithm for constructing a reliable mcds in probabilistic wireless networks. In Yu Cheng, Do Young Eun, Zhiguang Qin, Min Song, and Kai Xing, editors, *WASA*, volume 6843 of *Lecture Notes in Computer Science*, pages 96–107. Springer, 2011. ISBN 978-3-642-23489-7. 28

[21] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd HICSS*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society. ix, 5

[22] Alexander Hinneburg, Er Hinneburg, and Daniel A. Keim. An efficient approach to clustering in large multimedia databases with noise. pages 58–65. AAAI Press, 1998. 51

[23] J.L. Hurink and T. Nieberg. Approximating minimum independent dominating sets in wireless networks, 2007. URL `http://doc.utwente.nl/57741/`. 36

[24] O. Ilaya, C. Bil, and M. Evans. A particle swarm optimisation approach to graph permutations. In *Information, Decision and Control, 2007. IDC '07*, pages 366–371, 2007. ISBN 1-4244-0902. doi: 10.1109/IDC.2007.374578. 34

[25] Kamrul Islam, Selim G. Akl, and Henk Meijer. Maximizing the lifetime of wireless sensor networks through domatic partition. In *IEEE 34th Conference on Local Computer Networks*, pages 436–442, Zurich, Switzerland, Oct 2009. IEEE Xplore. ix, 3

[26] J.-P. Jamont, M. Occello, and A.Lagreze. A multiagent approach to manage communication in wireless instrumentation systems. *Measurement*, 43(4):489–503, Dec 2009. ix, 27, 56, 73

[27] Chunlin Ji, Yangyang Zhang, Shixing Gao, Ping Yuan, and Zhe Li. Particle swarm optimization for mobile ad hoc networks clustering. In *IEEE International Conference on Networking, Sensing and Control*, volume 1, pages 372–375, 2004. doi: 10.1109/ICNSC.2004.1297465. 34

[28] Junseok Kim, Sookhyeon Chang, and Younggoo Kwon. Odtpc: On-demand transmission power control for wireless sensor networks. In *International Conference on Information Networking, ICOIN*, pages 1–5, Jan 2008. ISBN 978-89-960761-1-7. 5

[29] R. V. Kulkarni and G. K. Venayagamoorthy. Particle swarm optimization in wireless-sensor networks: A brief survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(2):262–267, 2011. ISSN 1094-6977. doi: 10.1109/TSMCC.2010.2054080. 34

[30] Marco Mamei, Ronaldo Menezes, robert Tolksdorf, and Franco Zambonelli. Case studies for self-organization in computer science. *Journal of Systems Architecture*, 52(8):443–460, April 2006. ix

[31] Bernard Mans and Nirisha Shrestha. Performance evaluation of approximation algorithms for multipoint relay selection. In *Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop*, pages 480–491, 2004. 1

[32] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM. ISBN 1-58113-142-9. ix, 1

[33] Tim Nieberg and Johann Hurink. Wireless communication graphs. In *Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference*, pages 367–372, Dec 2004. ISBN 0-7803-8894-1. 2, 3, 11

[34] S. Olariu and A. Zomaya Q. XU. An energy-efficient self-organization protocol for wireless sensor networks. pages 55–60, Dec 2004. 4

[35] J. Guadalupe Olascuaga-Cabrera, Ernesto Lopez-Mellado, and Felix Ramos-Corchado. Self-organization of mobile devices networks. In *IEEE International Conference on System of Systems Engineering*, pages 1–6. IEEE, May 2009. ISBN 978-1-4244-4766-4. 14

[36] J.G. Olascuaga-Cabrera, E. López-Mellado, A. Mendez-Vazquez, and F.F. Ramos-Corchado. A self-organization algorithm for robust networking of wireless devices. *Sensors Journal, IEEE*, 11(3):771–780, march 2011. ISSN 1530-437X. doi: 10.1109/JSEN.2010.2068286. 35

[37] Konstantinos E. Parsopoulos and Michael N. Vrahatis. *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2010. ISBN 1615206663, 9781615206667. 40, 41, 42

[38] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report RR-3898. INRIA, 2000. ix, 2

[39] H Raei, M. Sarram, B. Salimi, and F. Adibniya. Energy-aware distributed algorithm for virtual backbone in wireless sensor networks. In *International Conference on Innovations in Information Technology.*, pages 435–439. Dec 2008. ISBN 978-1-4244-3396-4. 57

[40] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, NY, 1999. 41

[41] My T. Thai, Feng Wang, Dan Liu, Shiwei Zhu, and Ding-Zhu Du. Connected dominating sets in wireless networks with different transmission ranges. *IEEE Transactions on Mobile Computing*, 6(7):721–730, July 2007. 5

[42] J. Tillett, R. Rao, and F. Sahin. Cluster-head identification in ad hoc sensor networks using particle swarm optimization. In *IEEE International Conference on Personal Wireless Communications*, pages 201–205, 2002. doi: 10.1109/ICPWC.2002.1177277. 34

[43] Ozan K. Tonguz, Nawaporn Wisitpongphan, Jayendra S. Parikht, and Fan Bait. On the broadcast storm problem in ad hoc wireless networks. In *3rd International Conference on Broadband Communications, Networks and Systems*, pages 1–11, Oct 2006. ISBN 978-1-4244-0425-4. ix, 1, 6

[44] Laurent Viennot. Complexity results on election of multipoint relays in wireless networks. In *wireless networks, Report RR-3584, INRIA*, 1998. ix, 2

[45] Alec Woo and David Culler. Evaluation of efficient link reliability estimators for low-power wireless networks. Technical Report UCB/CSD-03-1270, EECS Department, University of California, Berkeley, 2003. URL `http://www.eecs.berkeley.edu/Pubs/TechRpts/2003/6239.html`. 29

[46] Xiaoling Wu, Shu Lei, Wang Jin, Jinsung Cho, and Sungyoung Lee. Energy-efficient deployment of mobile sensor networks by pso. In Heng Shen, Jinbao Li, Minglu Li, Jun Ni, and Wei Wang, editors, *Advanced Web and Network Technologies, and Applications*, volume 3842 of *Lecture Notes in Computer Science*, pages 373–382. Springer Berlin / Heidelberg, 2006. 34

[47] Bin Yang, Jinwu Xu, Jianhong Yang, and Debin Yang. A novel weighted clustering algorithm in mobile ad hoc networks using discrete particle swarm optimization (dpsowca). *Int. J. Netw. Manag.*, 20:71–84, March 2010. ISSN 1099-1190. doi: http://dx.doi.org/10.1002/nem.730. URL `http://dx.doi.org/10.1002/nem.730`. 34

[48] Ossama Younis and Sonia Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *INFOCOM. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, pages 629–640, March 2004. ISBN 0-7803-8355-9. ix, 5

[49] Ossama Younis, Student Member, and Sonia Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3:366–379, 2004. ix, 5

[50] Wen-Liang Zhong, Jian Huang, and Jun Zhang. A novel particle swarm optimization for the steiner tree problem in graphs. In *IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*, pages 2460–2467, 2008. doi: 10.1109/CEC.2008.4631127. 34

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Estrategias Locales para Formación y Mantenimiento de Redes Inalámbricas. Localized Strategies for Wireless Network Formation and Maintenance.

del (la) C.

J. Guadalupe OLASCUAGA CABRERA

el día 19 de Abril de 2013.

Dr. Luis Ernesto López Mellado
Investigador CINVESTAV 3C
CINVESTAV Unidad Guadalajara

Dr. Félix Francisco Ramos Corchado
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Ramón Parra Michel
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González Pico
Investigador CINVESTAV 2C
CINVESTAV Unidad Guadalajara

Dr. Andrés Méndez Vázquez
Investigador CINVESTAV Guadalajara 2C
CINVESTAV Guadalajara

Dr. Hugo Cesar Coyote Estrada
Gerente de Desarrollo
ORACLE