CT-880-SS1
DOU: 2015

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

# Optimización Multi-Objetivo para calidad de servicio en redes inalámbricas IEEE 802.11

Tesis que presenta:

**Arturo Raymundo Avilés**

para obtener el grado de:

**Doctor en Ciencias**

en la especialidad de:

**Ingeniería Eléctrica**



Directores de Tesis

**Dr. Mario Angel Siller González Pico**
**Dr Andrés Méndez Vázquez**

CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, Diciembre de 2014.

# Optimización Multi-Objetivo para calidad de servicio en redes inalámbricas IEEE 802.11

**Tesis de Doctorado en Ciencias**
**Ingeniería Eléctrica**

Por:
**Arturo Raymundo Avilés**
Maestro en Ciencias en la especialidad de Computación
Cinvestav Unidad Guadalajara 2007-2009

Becario de Conacyt, expediente no. 212737

Directores de Tesis
**Dr. Mario Angel Siller González Pico**
**Dr. Andrés Méndez Vázquez**

CINVESTAV del IPN Unidad Guadalajara, Diciembre de 2014

Centro de Investigación y de Estudios Avanzados

del Instituto Politécnico Nacional

Unidad Guadalajara

# Multi-Objective Optimization for Quality of Service in IEEE 802.11 wireless networks

A thesis presented by:
**Arturo Raymundo Avilés**

to obtain the degree of:
**Doctor of Sciences**

in the subject of:
**Electrical Engineering**

Thesis Advisors:
**Dr. Mario Angel Siller González Pico**
**Dr. Andrés Méndez Vázquez**

**CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, December 2014.**

# Multi-Objective Optimization for Quality of Service in IEEE 802.11 wireless networks

**Doctor of Sciences Thesis**
**in Electrical Engineering**

By·
**Arturo Raymundo Avilés**
Master of Sciences in Electrical Engineering
CINVESTAV 2007-2009

Scholarship granted by CONACYT, No. 212737

Thesis Advisors:
**Dr. Mario Angel Siller González Pico**
**Dr. Andrés Méndez Vázquez**

**CINVESTAV del IPN Unidad Guadalajara, December 2014.**

# Contents


**List of Figures** 5

**List of Tables** 7

**Abstract** 9

**Resumen** 9

**1 Introduction** 15
- 1.1 Index Terms 15
- 1.2 Motivation 17
- 1.3 Organization 18

**2 Literature Review** 19
- 2.1 The OSI Model 19
- 2.2 IEEE 802.11 20
- 2.3 Quality of Service 21
- 2.4 Cross-Layer Design Optimization 23
- 2.5 Optimization theory related works in 802.11 25
- 2.6 Multi-Objective Optimization 26
- 2.7 PSO for Multi-Objective Optimization 27

**3 IEEE 802.11 Optimization Model** 33
- 3.1 Network Model 33
- 3.2 Physical layer 33
- 3.3 Signal to Interference plus Noise Ratio 35
- 3.4 Bit Error Rate 36
- 3.5 Selective-Repeat Model 36
- 3.6 CSMA/CA Protocol 38

**4 IEEE 802.11 Mutil-Objective Solution** **41**
4.1 The Crossing Variables 42
4.2 Muti-Objective Model 43
4.3 Constraints definition 43
4.4 PSO Algorithm 44

**5 Experimental Results** **49**
5.1 NS-2 Validation 54

**6 Conclusions and Future Work** **59**
6.1 Contributions 59

**A Particle Swarm Optimization matlab Code** **61**

# List of Figures


1.1 Fields of interseccion of the tesis. 18

2.1 The first 3 layers of the OSI model. 20
2.2 Classification of main CLD works. 23
2.3 The PSO Algorithm. 28
2.4 Examples of different areas that the PSO has been used elaborated by [ ] 29

3.1 Snapshot of 802.11 PHY standard activities. 34
3.2 Snapshot of 802.11 MAC standard activities. 38

4.1 The used PSO Algorithm. 45

5.1 Framework. 50
5.2 Initial sampling of the particles 50
5.3 Results after few iterations. 52
5.4 Results of the PSO algorithm are shown the Pareto frontier of the physical layer 53
5.5 Results of the PSO algorithm are shown in the Pareto frontier of the protocol CSMA. 53
5.6 Results of the PSO algorithm are shown in the Pareto frontier of the protocol ARQ. 54
5.7 The evolution of the error. 55
5.8 Simulation in NS2 with 100 nodes 56
5.9 Results with 1000 nodes. 56
5.10 Results with 1000 nodes. 57
5.11 Results with 100 nodes. 57
5.12 Results with 100 nodes. 58

# List of Tables


5.1 PSO Control Variable Settings 51
5.2 Results only with the physical layer 52
5.3 Comparative between Standard and PSO with 1000 nodes 55
5.4 Comparative between Standard and PSO with 100 nodes 58

# Abstract

In this thesis, the end-to-end rate optimization problem in the 802.11 wireless network protocol is addressed with the use of Multi-Objective Optimization (MOO), Cross-Layer Design (CLD) and Particle Swarm Optimization (PSO).

The approach includes getting the cost functions of the first two layers of the Open Systems Interconnection model (OSI model). These are the physical layer and the Data link layer with its corresponding sub-layers Logical link control sub-layer (LLC) and Media access control sub-layer (MAC). With these cost functions we use Optimization Theory to improve network performance and optimize the throughput of the Network. The Particle Swarm Optimization algorithm (PSO) is an algorithm extensively used in various fields of science that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

PSO optimizes a problem by having a population of candidate solutions, called particles, and moving these particles around in the search-space according to a mathematical formula over the particle's position and velocity.

Additionally, the obtained results by the PSO, are verified using the network simulator (NS-2), Relevant changes are made for the implementations of the proposed model and a comparison with the original model.

## Resumen

Esta tesis tiene como objetivo ayudar a optimizar la entrega de paquetes fin a fin, haciendo uso de la teoría de optimización. Las redes inalámbricas se caracterizan por el uso de un medio de transmisión dinámico, en el que las condiciones de transmisión cambian continuamente. Esta tesis trabaja con el protocolo IEEE 802.11 (WiFi).

El trabajo incluye obtener las funciones de costo de las 2 primeras capas del modelo de interconexión de sistemas abiertos (modelo OSI) incluyendo la capa física y la capa de enlace de datos con sus subcapas control de enlace lógico (LLC) y subcapa de control de acceso al medio (MAC). Con estas funciones de costo hacemos uso de la teoría de optimización, para optimizar el rendimiento de la red. El algoritmo de optimización de enjambre de partículas (Particle Swarm Optimization, PSO) es un algoritmo de optimización heurístico que optimiza el problema iterativamente tratando de mejorar una solución candidato con respecto a una determinada medida de la calidad. PSO optimiza un problema al tener una población de soluciones candidatas (partículas) y moviéndose alrededor de estas partículas.

Después para comprobar los resultados obtenidos por el PSO, se hace uso del simulador de redes NS-2, donde se hacen las modificaciones pertinentes en el simulador para realizar las implementaciones del modelo original y del modelo propuesto.

# Dedication

First and foremost I would like to thank God.

I also thank my parents and sisters, who have offered me their unconditional in every stage of my life support.

Thanks to CINVESTAV and all its people: Companions, Don Pedro, cops, maintenance personnel, and academic library, for giving us their hospitality.

To my thesis advisor Dr. Mario Angel Gonzalez Pico Siller thanks for the support, time and dedication in the preparation of this thesis.

My co-advisor Dr. Andres Mendez Vazquez for their patience and great support at meetings that were necessary in order to complete this work successfully.

Finally I want agraecer the most important person in my life Dr. Mayra Trejo Hernandez, Thank you always being there for me.

This research work has been possible thanks to financial support of CONACYT that provided my scholarship.

# Chapter 1

# Introduction

## 1.1 Index Terms

Multi-Objective Optimization (MOO), Cross-Layer Design (CLD), Particle Swarm Optimization (PSO), Open Systems Interconnection model (OSI model), Logical link control sub-layer (LLC), Quality of Service QoS, (CSMA/CA), Automatic Repeat Request protocol (ARQ), Hybrid Coordination Function (HCF), Simple Multi-Objective Particle Swarm Optimizer (SMOPSO), Distributed Coordination Function DCF, Time Division Multiple Access (TDM), Multi-Objective Optimization Problem (MOOP), Quadrature Amplitude Modulation (QAM), Signal Noise Ratio (SNR), Signal to Interference plus Noise Ratio (SINR), Bit Error Rate (BER), Crossing Variables (CV), Packet Success Rate (PSR), Media access control sub-layer (MAC), Network Simulator 2 (NS-2), Tcl (Tool Command Language, physical layer (PHY), Wireless LAN (WLAN).

The IEEE 802.11 standard is one of the most developed wireless technologies and it plays a major role in communication networks around the world [ ], [ ]. The main attributes of the 802.11 networks are its flexibility, simplicity, cost effectiveness, and others. This technology provides people with a ubiquitous communication in computing environments: in offices, hospitals, universities, factories, airports, almost anywhere. Simultaneously, multimedia applications have experienced an explosive growth. More than ever, people are now requiring high-speed video, audio, voice and above all, web services. This is true even when they are moving in their offices or universities using a dynamic transmission medium, in which transmission conditions change continuously. For this, multimedia applications require a certain level of Quality of Service (QoS) support such as: guaranteed bandwidth, guaranteed delay, minimum jitter and transmision error. Providing these QoS requirements, in the 802.11 networks is still a challenge due to the QoS-unaware functions of its Medium Access Control (MAC) sub-layer, the noisy and variable physical (PHY) layer characteristics and variable traffic load conditions.

Therefore, QoS is a key problem for today's IP networks. Many frameworks (IntServ, DiffServ, MPLS, etc.) have been proposed to provide service differentiation in the Internet [ ], [ ]. At the same time, the Internet is becoming a heterogeneous medium due to the recent explosion in the use of wireless networks. In addition, in wireless environments, bandwidth is scarce and channel conditions are time-varying and sometimes highly loss. Although IEEE 802.11 Wireless LAN (WLAN) is the most widely used WLAN standard today, it cannot provide QoS support for the increasing number of multimedia applications. Thus, a large number of 802.11 QoS enhancement schemes have been proposed, each one focusing on a particular mode.

For example, a possible way to overcome problems of QoS is the use of cross layer optimization, which has become a very active research area over the last few years [ , ]. Although, there are several works that implement diverse algorithms to improve the Quality of Service (QoS) [ ], there is still a lot of development that could be done to improve network bandwidth, reliability and quality of the networks in general.

Moreover, nowadays, traffic carried by wireless networks is a mix of real-time traffic such as voice, multimedia conferences, games, web browsing, messaging and file transfer. All of these applications require varying and diverse standards of QoS because it is desirable to obtain the best throughput under different types of traffic. Additionally, this topic has received significant attention from both academy and industry over the last few years. [ ]. For example, one of the main problems is the scheduling task, which is motivated by the unique features of wireless networks: scarce resources, mobile users, interference from other users in the network, and time-varying channel conditions (due to mobility). Therefore, good scheduling schemes in wireless networks should opportunistically seek to exploit channel conditions to achieve better network performance. Another problem is the multi-hop in wireless networks.

In addition to the previous problems, other difficulties that need to be addressed are: How should one determine the end-to-end data rates for the users? Or when should a given link be activated in the network? How can one ensure that the rate provided by the links is enough to support the end-to-end rate of all users? Perhaps, most importantly, can we develop efficient distributed solutions to these problems?

In addition to all these questions, the wireless medium is a multi-access medium, where the users' transmissions interfere with each other, and where the channel capacity is time-varying due to user's mobility, multi-path, etc. This causes dependencies across users and network layers that simply are not present in their wired counterpart. Despite such difficulties, there have been significant advances that demonstrate that wireless resources across multiple layers, such as frequency, power, link data rates and end-user data rates, can be incorporated into a unified optimization framework [ ].

For example, in [ ] R. Madan et al. present a technique to maximize the lifetime of the network using a convex optimization. In addition, they proposed an iterative algorithm which alternates between adaptive link scheduling, computation of optimal link rates and transmission powers for a fixed link schedule. In [ ] Yayu Gao et al. presented a throughput analysis of an M-group IEEE 802.11 DCF network. For this, they considered a noiseless channel and assumed that the buffer size of each node and the maximum number of retransmission attempts of each head-of-line packet are infinite. Clearly, this constraint was not very realistic, and it is a drawback at the time of implementation.

On the other hand, the MOO optimization framework has been used successfully in trying to model the multilayer structure of the network protocols [ ]. For this reason, this work presents a method for optimizing the throughput and delay of the networks 802.11 with the use of MOO algorithms. For this purpose, a model of the physical MAC and Logical Link Control (LLC) layers is used to model the performance of the network. Then, PSO is used to obtain the global solutions. Three case studies were proposed to achieve this, and the non-dominated solutions [ ] were taken into account.

## 1.2 Motivation

As previously mentioned, the 802.11 protocol is a very popular, and the main motivation in this work is to evaluate the intersection between the fields of optimization theory and networking area, see (Fig. 1.1). There are a very few studies in which these two fields converge. The aim of this thesis is develop a 802.11 performance model from which the features to be used in an optimization algorithm derived, to obatain the results using a network simulator.

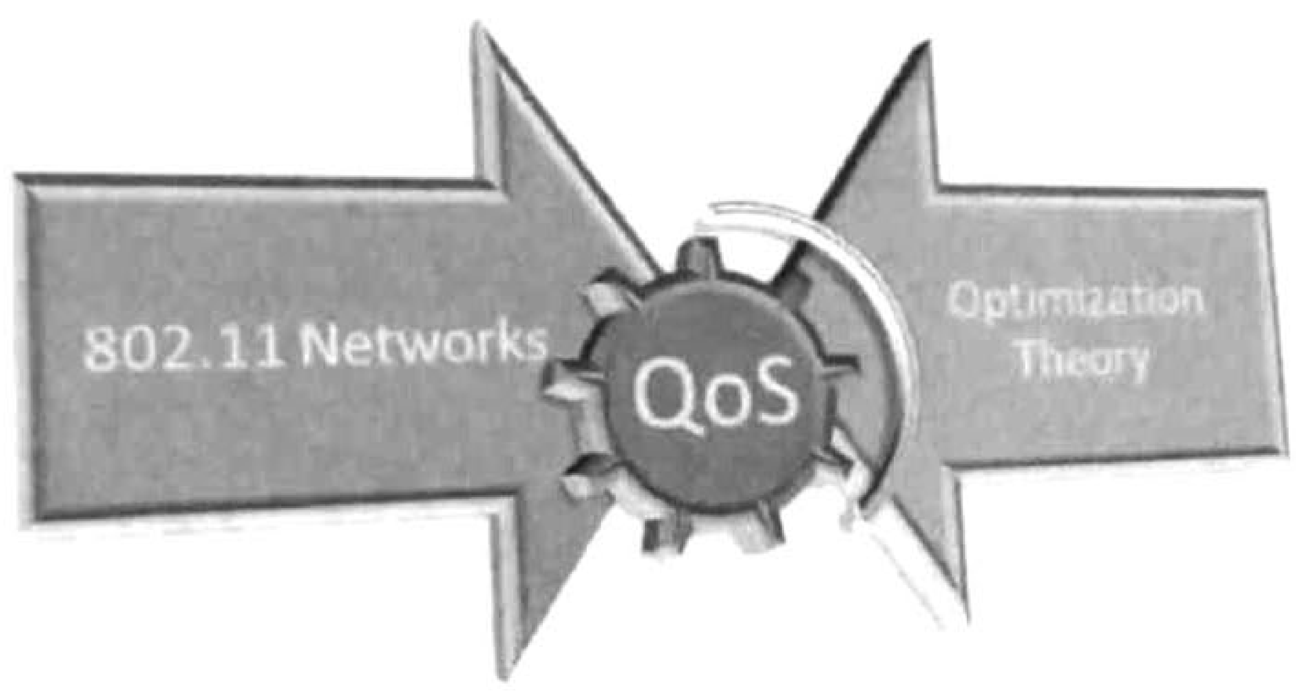


Figure 1.1: Fields of interseccion of the tesis.

## 1.3 Organization

The remainder of this thesis is organized as follows, (Chapter. 2) includes a brief description of the theoretical framework: the OSI Model, (Section 2.1), the standard 802.11 (Section 2.2), Quality of Service (Section 2.3), Cross-Layer Design (Section 2.4), some related works in (Section 2.5) and Multi-Objective Optimization (Section 2.6).

(Chapter 3) provides the definition of the Multi-Objective functions, that are divided as follows: (Section 3.2) it explains in detail de physical layer; in (Section 3.3) the Signal to Interference plus Noise Ratio is explained; in (Section 3.4) the bit error rate is present; in (Section 3.5) the automatic repeat request protocol is described and in the last section of this chapter (Section 3.6), an explanation of the media access control is given.

In (Chapter 4), the proposed Multi-Objective model is described. Next, simulation results and a benchmark performance of the proposed algorithm are discussed in (Chapter 5). Finally, concluding remarks are given in (Chapter 6).

# Chapter 2

# Literature Review

The study of the state of the art will be explained in the following five subsections: The OSI model (Sec. 2.1), the standard IEEE 802.11 (Sec. 2.2), QoS (Sec. 2.3), CLD (Sec. 2.4), some works of optimization in wireless networks (Sec. 2.5) and Multi-Objective Optimization (Sec. 2.6).

## 2.1 The OSI Model

The model that is going to be used through this thesis is based in the Open Systems Interconnection model (OSI model), which is a product of the Open Systems Interconnection effort at the International Organization for Standardization. Specifically, the model is taking as reference the physical layer and data link layers [ ] described in 802.11. This is a description of characterized and standardized the functions of a communication system in terms of abstraction layers. An example of the first layers of the OSI model can be seen in (Fig. 2.1), and a general descriptions of which task are performed at each layer is given in the following paragraphs.

- Physical layer, in which data is handled in the form of bits, and the performed functions are: bit-by-bit or symbol-by-symbol delivery, the modulation and multiplexing.
- Data link layer, in which the data is in the form of frames. It uses the packet success rate the bit error rate, average number of frames sent and the probability of sending a frame, through a two sub-layer architecture, LLC sublayer manages in the Error control by using the

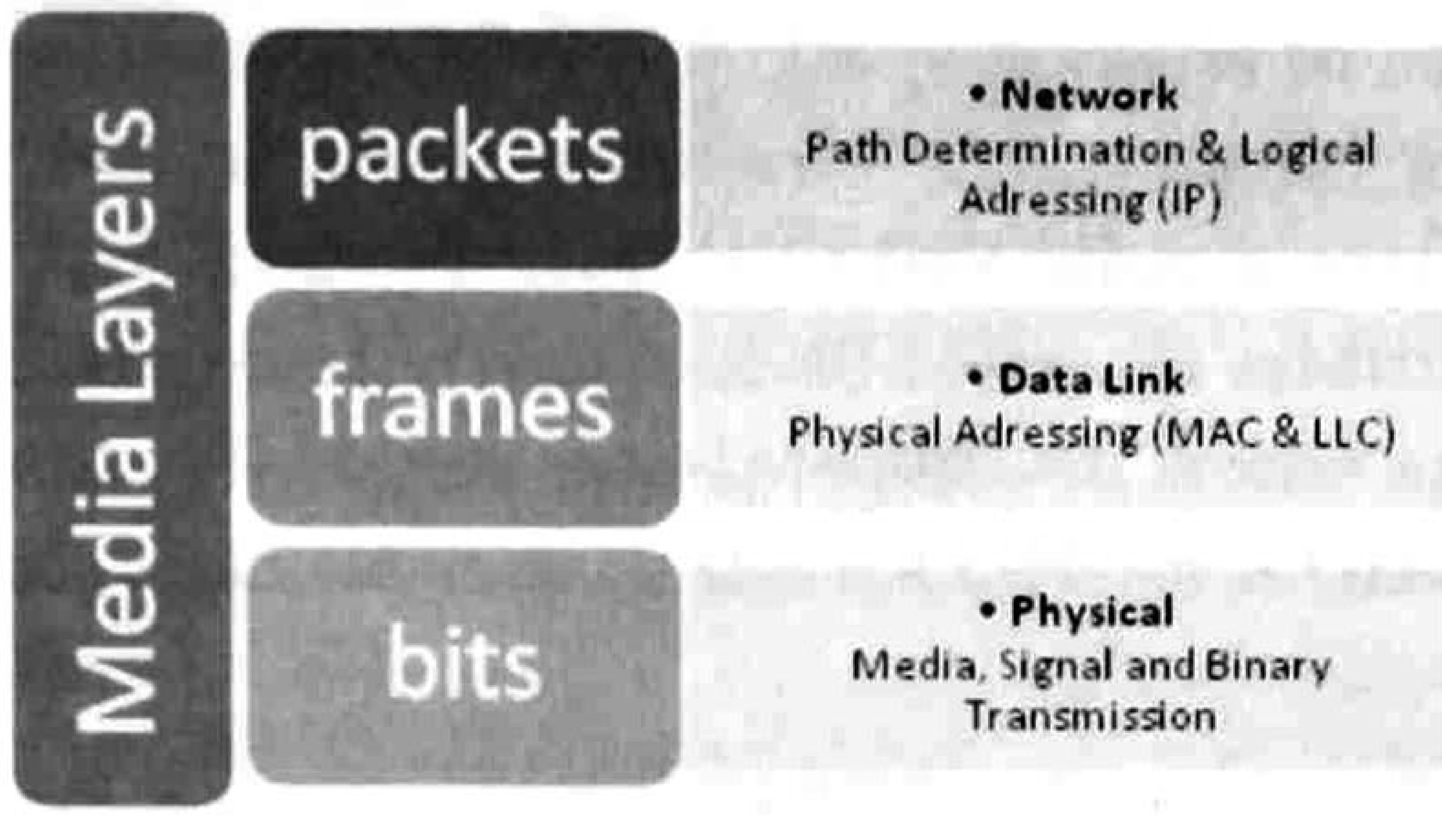


Figure 2.1: The first 3 layers of the OSI model.

Automatic Repeat Request (ARQ) protocol. The MAC sublayer focuses in Multiple access protocols (CSMA/CA) and QoS control.

Finally, because of the evolution of the IEEE 802.11, an overview of the protocol is given in the following section.

## 2.2 IEEE 802.11

Over the last decade the protocol IEEE 802.11 has become the most widely used protocol around the world. IEEE 802.11 is a set of physical layer and medium access control specifications for implementing wireless local area network computer communication. They were created and maintained by the IEEE Standards Committee (IEEE 802). The original version of the standard IEEE 802.11 was released in 1997 and clarified in 1999. It specifies network bit rates of 1 or 2 megabits per second (Mbit/s), plus forward error correction code [ ]. In addition, it specifies a three alternative physical layer architectures: a diffuse infrared, operating at 1 Mbit/s; a frequency-hopping spread spectrum, operating at 1 Mbit/s or 2 Mbit/s; finally, a direct-sequence spread spectrum, operating at 1 Mbit/s or 2 Mbit/s. The two latest radio technologies use microwave transmission over the Industrial Scientific Medical frequency band at 2.4 GHz. Some earlier WLAN technologies used lower frequencies, such as the U.S. 900 MHz ISM band.

The proposed Multi-Objective model uses as a basis the IEEE 802.11 [ ] and IEEE 802.11n [ ] protocols. The IEEE 802.11n is a wireless networking standard that uses multiple antennas to increase data rates, and it is one of the most used versions around the world. It is an amendment of the IEEE

802.11-2007 wireless networking standard [ ]. Its purpose is to improve network throughput over the two previous standards with a significant increase in the maximum net data rate from 54 Mbit/s to 600 Mbit/s with the use of four spatial streams at a channel bandwidth of 40 MHz. In addition 802.11n standardizes support for multiple-input, multiple-output, frame aggregation, security improvements, and it can be used in the 2.4 GHz or 5 GHz frequency bands, among other features.

## 2.3 Quality of Service

QoS is becoming an increasingly important element of any communications system, and refers to several related aspects of computer networks that allow the transport of traffic with special requirements, for instance providing a consistent and predictable data delivery service [ ].

The IEEE 802.11 QoS facility provides MAC enhancements to support LAN applications with QoS requirements. The QoS enhancements are available to QoS stations associated with a QoS access point in a QoS BSS [ ]. A subset of the QoS enhancements is available for use between a set of stations (STAs) which are members of the same QoS IBSS. Similarly, a subset of the QoS enhancements is available for use between neighbor peer mesh STAs. In wireless networks, the capacity of each link depends on the signal and interference levels. Thus, it depends on the power and transmission schedule at other links. This relationship between the link capacity, power assignment, and the transmission schedule is typically non-convex [ ].

Essentially, all this information is saying that in networks, QoS is affected by various factors, which can be divided into human and technical factors. Human factors include: stability of service, availability of service, delays, user information. Technical factors include: reliability, scalability, effectiveness, maintainability and grade of service.

QoS refers to meeting traffic transmission special requirements, for instance providing a consistent and predictable data delivery service. The IEEE 802.11e/g QoS facility provides MAC enhancements to support LAN applications with QoS requirements. However, the initial IEEE 802.11 did not support QoS and several challenges had to be addressed as a consequence. Some of the issues that have been studied include: service differentiation in the MAC layer, admission control and bandwidth reservation in MAC and higher layers, and link adaptation in the physical layer.

For instance, Zhai et al. [ ] studied how well the 802.11 protocol support QoS. They analyzed packet delay and throughput under non-saturation conditions. The results showed that if the network is tuned to work at optimal point it is then possible to achieve maximum throughput, low delay and packet loss rate. A drawback in their work is that they assume that the traffic is uniformly distributed among the nodes, that in the real life this assumption is not very useful.

In [ ] Stefan Mangold et al. they analyzed the IEEE 802.11e protocol for QoS support they mentioned that the medium access control protocol is the standard for wireless networks to provide QoS. They analyzed possible enhancements as the Hybrid Coordination Function (HCF) for QoS support in 802.11e, and they compared its performance to the legacy 802.11 standard. A drawback in their work is that they used event-driven stochastic simulations to evaluate the performance of the MAC and the physical layer, which limited the simulation size of the network.

Another interesting paper of modeling is [ ] in which Bianchi proposed a system model of Markov chain to analyze and optimize the performance of IEEE 802.11 DCF and this model is extensively in works of modeling in the literature (Most of them focused on homogeneous IEEE 802.11 DCF networks) he prove that some stations suffer of throughput degradation when they access to the shared channel and when the load of the channel is high. He used a two-dimensional Markov chain of $m$ backoff stages in which each stage represents the backoff time counter of a node. A transition takes place upon collision and successful transmission, to a higher stage and return to the first stage. A drawback is that in networks, the nodes may have changing traffic rates and several QoS requirements.

In [ ] Felemban et al. they introduced two models for IEEE 802.11 DCF protocol in a single hop setting under both saturated and unsaturated traffic loads. They presented a precise analytical model of the DCF extended from Bianchi, which takes the freezing probability of the Backoff Counter into account, if the channel is sensed as busy. In their work they assumed that a general buffer network for each node with a buffer of infinite size. They took the freezing probability of the Backoff Counter into account, if the channel was sensed as busy.

In [ ] Pourmohammad et al. propose an analytical model to estimate QoS. Their model is based on the queuing theory and investigates the stochastic behavior of data transmission in wireless ad hoc networks. They include network layer processing time and advanced queue management schemes in

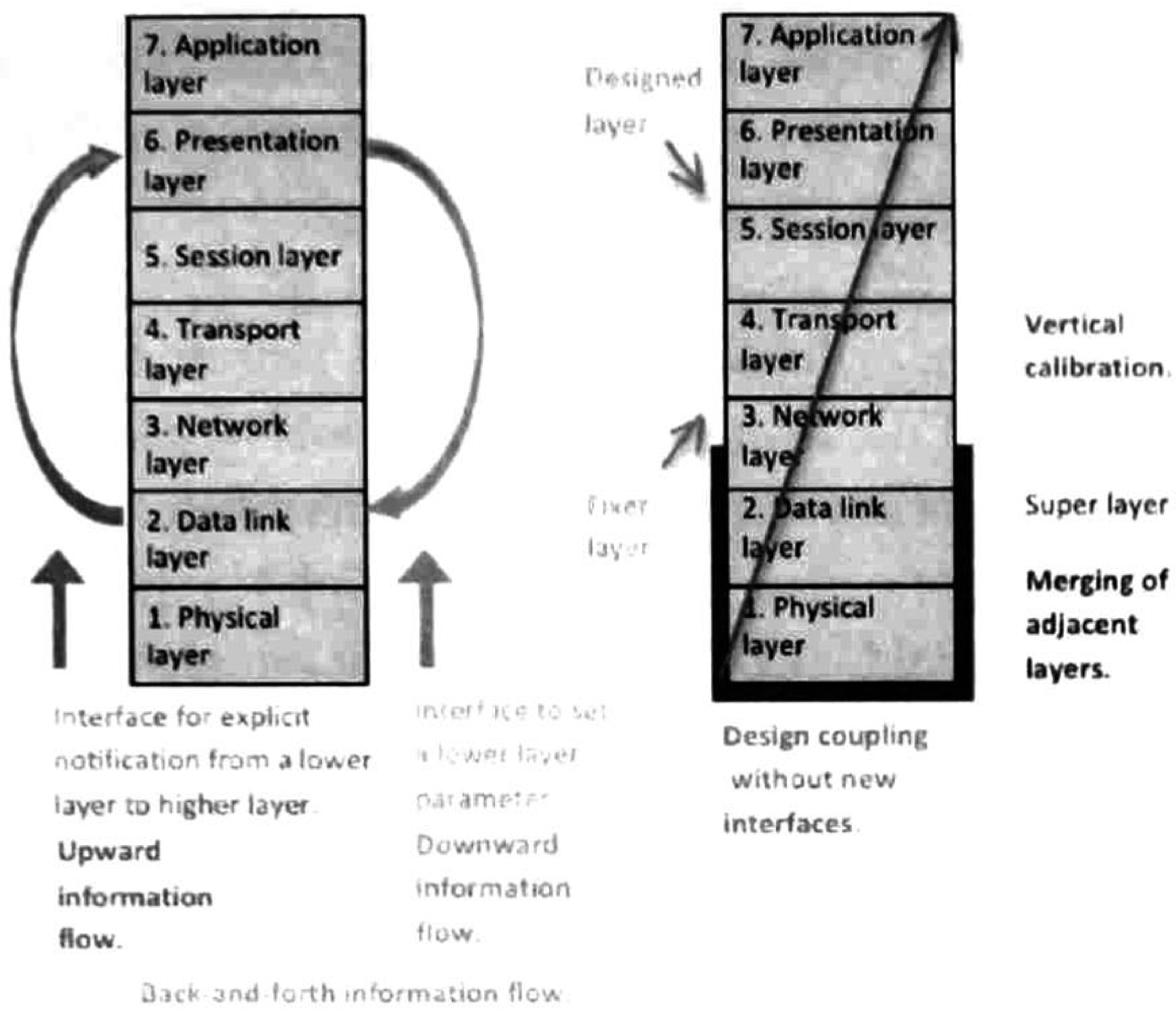


Figure 2.2: Classification of main CLD works.

the modeling process. A drawback is that their queue management scheme is not suitble for real life networks.

## 2.4 Cross-Layer Design Optimization

There have been a large number of studies on the use of the CLD methodology in order to optimize error rate, throughput, delay and latency. The most general Cross-Layer solutions for joint optimization congestion control and scheduling have recently been developed by several researchers [ ].

[Fig. 2.2] shows the most general ideas that are used for the implementation of the CLD. In the left OSI stack the first approach, in red, shows upward information flow. Basically data is taken from a lower layer to a higher one. The second one in green shows down forward information flow i.e. it takes data from a higher layer to a lower layer. Finally, the third is the combination of the previous two. In the right side of the design, three different approaches are shown: the first design couples, without new interfaces, a super layer and the third in brown vertical calibration, which avoids using any formal knowledge of the layers.

For example, according to [ ] the traffic in wireless networks is a mix of real time multimedia and non-real time file transfers. Vinet et al. mention that wireless networks misinterpret the packet loss due to collision as a route failure and this triggers a route maintenance phase. This causes unnecessary overhead resulting in low throughput. In their work, they present a service driven CLD model for the purpose of increasing the throughput by dynamically adjusting the limits of Request to Send retransmissions of the MAC sublayer for different flows in the network according to a given priority.

Next, in [ ], Choi et al. present a framework for analyzing complex Cross-Layer interactions in 802.11 WLANs. Although, it is well known that implementing a tool for correct performance measurement of Cross-Layer design is extremely difficult, the authors propose a framework for analyzing complex Cross-Layer interactions in 802.11 WLANs, with the aim of providing effective tools for understanding and improving WLAN performance. Their model is based on the Bianchis IEEE 802.11 DCF model [ ], and it is presented in (Eq. 2.1):

$$\tau_i = \frac{2\,(1-2p_i)}{(1-2p_i)\,(W_0+1)+p_i W_0\,(1-(2p_i)^m)}, \tag{2.1}$$

where the number $m$ is the maximum backoff stage and $W_0$ is the minimum backoff window size, and the conditional frame failure probability $p_i$.

Another example is given by Cui et al. [ ] where a joint routing, MAC, and link layer optimization are proposed to address the problem of the energy consumption in Sensor Networks. In this work, the authors consider a variable-length Time Division Multiple Access (TDMA) scheme and a Quadrature Amplitude Modulation (QAM) for the optimization problem. For this reason, the energy consumption cost function includes both transmission energy and circuit processing energy. Therefore, based on their analysis, it is shown that a single-hop communication could be the optimal in some cases in which the circuit energy dominates the energy consumption instead of the transmission energy. Although, the optimization problems presented in that paper are quite interesting, no communication protocol for practical implementation is proposed. Moreover, the issues at the transport layer, such as congestion and flow control, are not taken into consideration.

In [ ], Madan et al. proposes an optimization involving the transmission power, transmission rate, and link schedule for TDMA-based WSNs. In this work, the optimization is performed to maximize the network lifetime, instead of minimizing the total average power consumption. their work is based on convex optimization. In addition, they proposed an iterative algorithm that alternates between adaptive link scheduling, computation of optimal link rates, and transmission powers for a fixed link schedule.

In [ ], Xiaojun Lin et al. present a tutorial on Cross-Layer Optimization in wireless networks. They mention that optimization-based approaches have been used over the past several years to study resource allocation problems in communication networks. For example, wireless networks due to interference, require sophisticated scheduling mechanisms to carefully select only a subset of links to be activated at each time.

The scope of each link depends on the signal and interference levels, and on the power and transmission schedule at other links. We observed that the essential features of many wireless Cross-Layer control problems are not convex. It is deduced that convex programming is often not enough. In addition, the authors in [ ], mention that the optimal scheduling component at the MAC layer is very complex, and thus needs simpler distributed solutions. Therefore, the scheduling component needs to solve a difficult non-convex problem, and it usually becomes the bottleneck of the solution. This inherent non-convexity in the scheduling component requires advanced techniques, in addition to convex programming, to satisfactorily solve the Cross-Layer control problem in wireless networks.

## 2.5 Optimization theory related works in 802.11

There are many works in the area of optimization in wireless networks, but in general for the case of the standard 802.11 the works do not follow the according to the general rules of the optimization theory. By example in [ ], Laddomada et al. present a distributed coordination function (DCF) and focuses in optimizing the aggregated throughput. The main idea of their work derives from the statement that links aggregate throughput of the network to the packet rate of the contending stations. They based their work in the model from Bianchi [ ], however they do not present a real optimization model, since only makes modifications in the design.

## 2.6 Multi-Objective Optimization

MOO is an area of multiple criteria decision making that is concerned with mathematical optimization problems involving more than one objective function. MOO has been applied in many fields of science, where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. The Multi-Objective problems contain a collection of objective functions that might cause conflicts between their different goals because they must be satisfied at the same time. In this work, a MOO model is proposed to model a CLD of the first two layers of the 802.11 protocol.

A classical notation [ ] for the MOO is the following one:

$$F(X) = \{f_1(X), f_2(X), \ldots, f_n(X)\}, \tag{2.2}$$

subject to:

$$\begin{aligned} H(X) &= 0 \\ G(X) &\geq 0. \end{aligned} \tag{2.3}$$

In this case, the functions to be optimized are the set of functions $F(X)$, where the Vector $X$ is the set of independent variables. Functions $H(X)$ and $G(X)$ are the constraints of the model. The methods used in MOO provide solutions by using the idea of dominance (Def. 1 ).

**Definition 1** (Dominated). *A particle is called dominated if there exist a vector $\vec{x} = (x_1, \ldots, x_k)$ that is said to dominate another vector $\vec{v} = (v_1, \ldots, v_k)$ if and only if:*

$$\forall i \in \{1, \ldots, k\},\ x_i \leq v_i \wedge \exists i_1 \in \{1, \ldots, k\}\ |x_1 < v_1. \tag{2.4}$$

**Definition 2** (Non-dominated Set). *In a set of solutions, the non-dominated set of solutions $X'$ are those that are not dominated by any member of the set $X$.*

The main problem of MOO is the conflict that appears when the multiple functions are processed to obtain their optimal solutions. For example, this happens when there are two functions, one to be minimized and the other to be maximized. Because they share variables, classical methods of optimization tend to obtain feasible solutions, but they are not the optimal to fulfill the cost function goals and constraints.

Solving Multi-Objective optimization problems typically means finding the most preferred solution as the final one. The best solution is a Pareto optimal solution, which is the best option.

**Definition 3** (Pareto Optimal). *The main Pareto Optimal. A solution $x^*$ is Pareto Optimal if and only if there is not another vector $v = f(x)(v_1, \ldots, v_k)$ that dominates to $f(x^*)(u_1, \ldots, u_k)$.*

Osyczka [ ] et al. comment that the Multi-Objective Optimization Problem (MOOP) can be defined as the problem of finding a vector of decision variables that satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of the performance criteria which are usually in conflict with each other. Hence, the term optimize through finding a solution which would give the best values for all the objective functions acceptable to the decision maker. Whether or not solving a particular multi-objective optimization formulation serves as a necessary and/or a sufficient condition for Pareto optimality is central to its performance.

Although, classical optimization methods have been used to try to solve the Multi-Objective optimization problem, they tend to obtain local solutions. Therefore, it has been necessary to look for new optimization algorithms for the Multi-Objective problem. For example, evolutionary algorithms have shown that they are able to obtain global solutions over time [ ]. For example, a genetic algorithm-based Multi-Objective technique is presented in [ ], where multiple non-dominated solutions can be obtained in a single run. However, the optimization problem is considerably simplified. In addition, the proposed technique is computationally complex due to the ranking process during the fitness assignment procedure. Therefore, the need to look at more efficient techniques for optimization as the PSO is evident.

## 2.7 PSO for Multi-Objective Optimization

In computer science, optimization for Particle Swarm Optimization refers to a series of methods and heuristic optimization algorithms that evoke the behavior of swarms of bees in nature, (Fig. 2.3)

The PSO has been used in a variety of works along the literature [ ], [ ] where the algorithm is used to solve real problems and simulations of

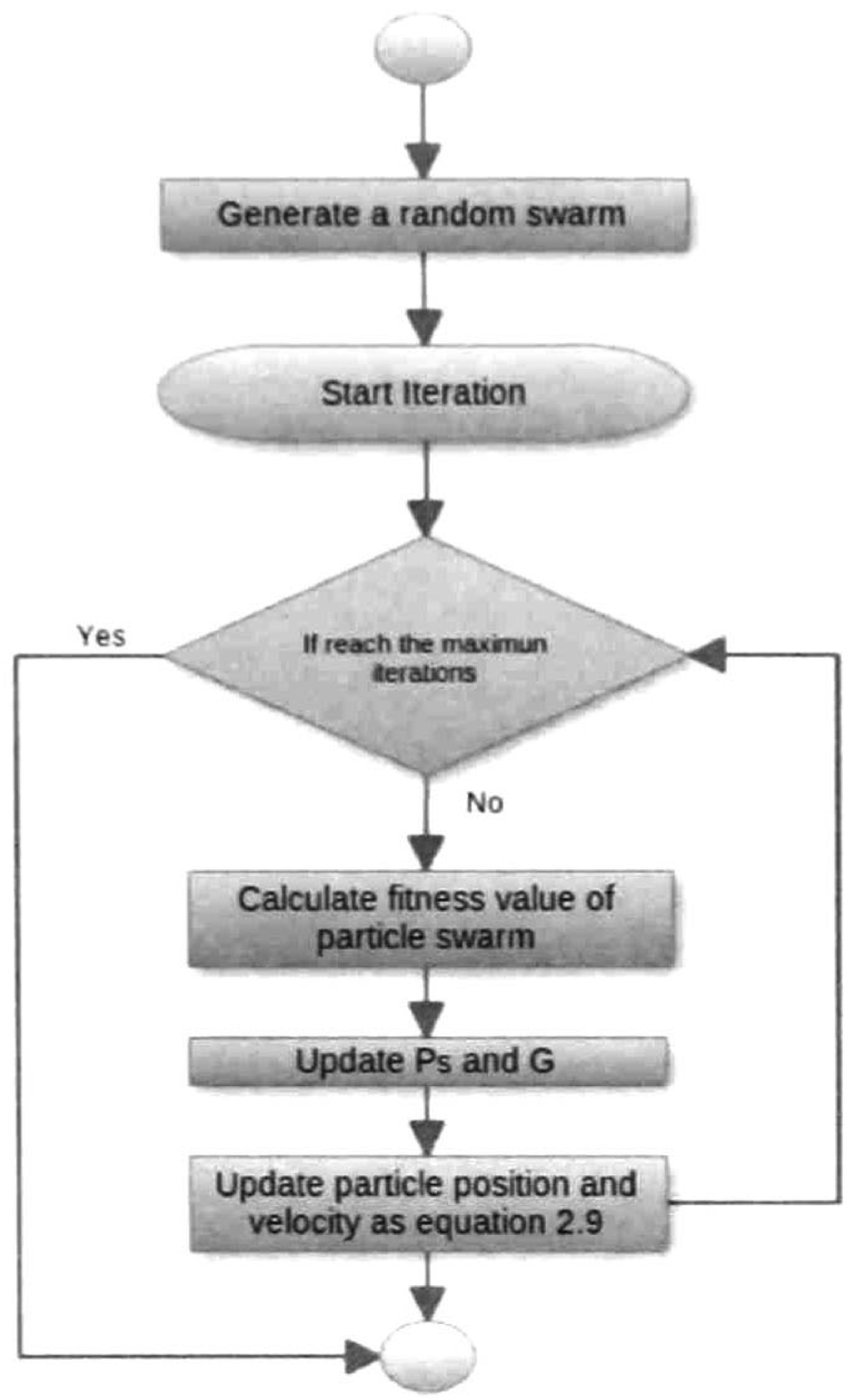


Figure 2.3: The PSO Algorithm.

test functions. For example, in [ ], the authors propose an approach called Simple Multi-Objective Particle Swarm Optimizer (SMOPSO) which incorporates an elitist policy and Pareto dominance. The authors test their algorithm against a series of equations and compare the results against the ones obtained by Pareto Archived Evolution Strategy (PAES)[ ] and the Multi-Objective Genetic Algorithm 2 (MOGA2) [ ].

First, in [ ], the authors define a vector $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ which satisfies the $m$ inequality constraints:

$$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \ldots, m. \tag{2.5}$$

In addition, the authors utilize $p$ equality constraints:

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p. \tag{2.6}$$

With all this, they define the optimization vector function as:

$$\vec{f}(\vec{x}) = [f_1(\vec{x}),\ f_2(\vec{x}),\ \ldots\ , f_k(\vec{x})]^T\,. \tag{2.7}$$

According to [ ] the constraints given in (Eq. 2.5) and (Eq. 2.6) defines the feasible region $\Omega$, then any point in $\Omega$ defines a feasible solution. The authors extend the classical PSO with the use of a Uniform Mutation Operator that selects one dimension of the particle with a certain probability, later changing its value. They use an elitist policy with the objective of maintaining the best solutions.

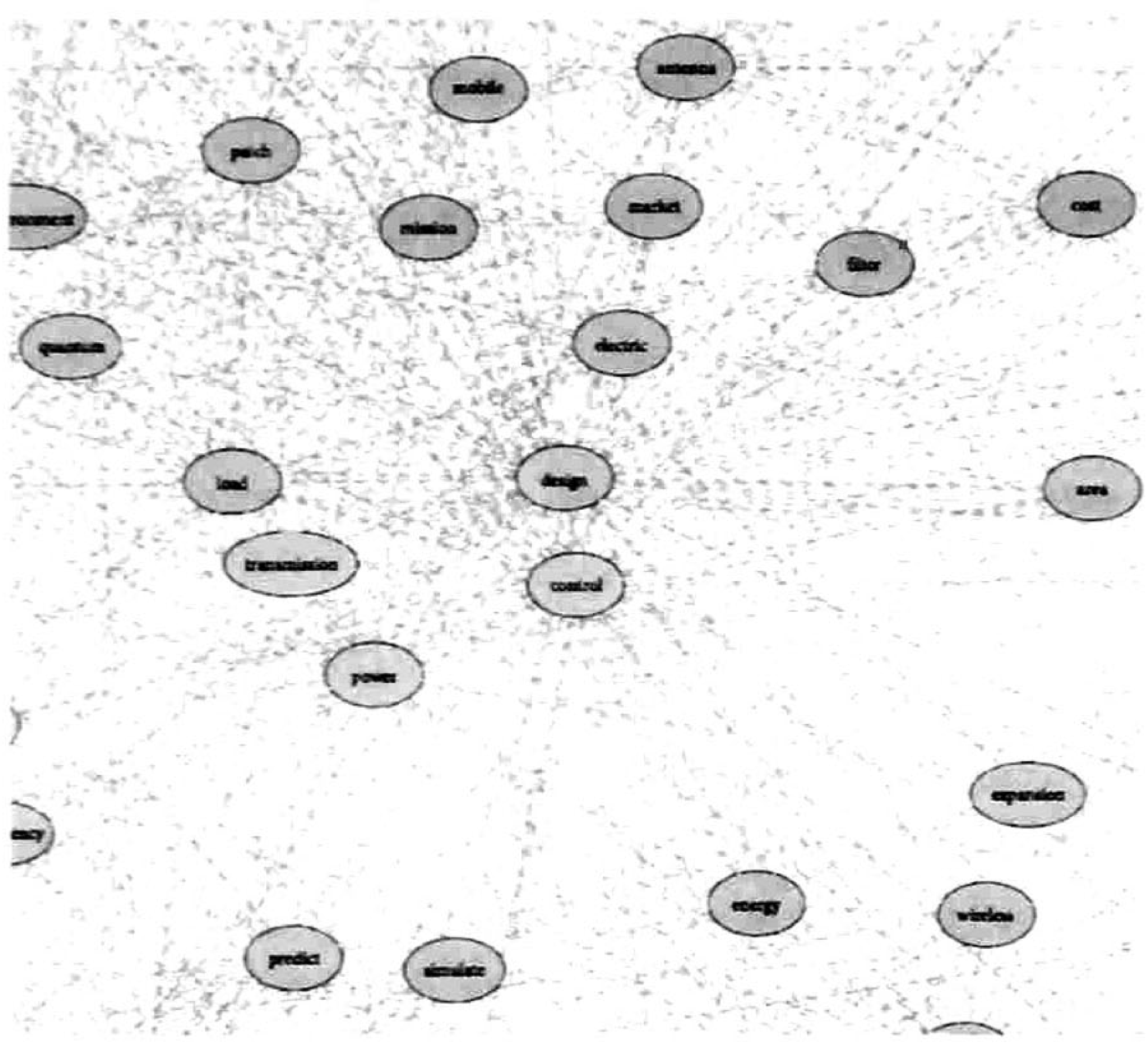

Figure 2.4: Examples of different areas that the PSO has been used elaborated by [ ].

Furthermore, Poli et al. [ ] show an analysis of the different areas where the PSO has been used: biomedical, communication networks, clustering and classification, control (one of the largest), distribution networks, financial, robotics, power systems, signal processing, etc. In addition, Poli et al. give a canonical version of the PSO, where each particle is moved by two elastic forces: one attracting force with random magnitude to the fittest location so far encountered by the particle, and the other attracting force with random magnitude to the best location encountered by any of the particle's social neighbor in the swarm. In addition, if the problem is $N$-dimensional, each particle position and velocity can be represented as a vector with $N$ components. Therefore, using the velocity vector, $v = (v_1, \ldots, v_N)$, which represents the velocity $v_i$ of each particle, is given by:

$$v_i\,(t+1) = \omega v_i(t) + \psi_1 R_1\,(x_{si} - x_i(t)) + \psi_2 R_2\,(x_{pi} - x_i\,(t))\,, \tag{2.8}$$

where $x_{si}$ is the ith component of the best point visited by the neighbors of the particle; $x_i(t)$ is the ith component of the particles current location; $x_{pi}$ is the ith component of its personal best; $R_1$ and $R_2$ are two independent random variables uniformly distributed in [0, 1]; $\omega$ is a constant known as the inertia weight; $\psi_1$ and $\psi_2$ are two constants, known as the acceleration coefficients, which control the relative proportion of cognition and social interaction in the swarm.

Another formula is the position of a particle (Eq. 2.9), which is updated every time step.

$$x_i(t+1) = x_i(t) + v_i(t+1). \tag{2.9}$$

The previous (Eq. 2.8) has been modified into the following version:

$$v_i\,(t+1) = \kappa\,(v_i(t) + \psi_1 R_1\,(x_{si} - x_i(t)) + \psi_2 R_2\,(x_{pi} - x_i(t)))\,, \tag{2.10}$$

where $\kappa$ is a constant called the constriction coefficient. Poli et al. mention that if $\psi_1$, $\psi_2$, and $\kappa$ are correctly chosen, the PSO is guaranteed to converge without the need for special constraints in the model.

The PSO algorithm works by having a population of candidate solutions. These solutions (particles) are moved around in the search-space according to a few simple formulas. The movements of the particles are guided by their own best-known position in the search-space as well as by the entire swarm's best known position. When improved positions are being discovered, these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

In PSO, the choice of parameters is a determining factor in the performance of the optimization algorithm. Therefore, one of the most important tasks is the selection of a set of parameters that promote a good performance of the algorithm. This is due to the fact that the PSO may easily find the local optimal solution instead of the global optimal solution [ ]. This premature convergence can be avoided by ignoring the best-known global position

$g$, and taking in place the best position $l$ known as sub-swarm "surroundings" This sub-swarm can be defined geometrically or in a social way, i.e. as a related $d$ regardless of the distances between particles.

Finally, the PSO algorithm (Algorithm 2.7) is an iterative one, and each particle moves through the fitness landscape at each iteration, according to its current fitness values as well as those of nearby particles, as well as the swarm as a whole. A pseudo-code for PSO implementatio is shown next.

**Algorithm 1** Particle Swarm Optimization Algorithm

```
PSO {
Initiate-Pop();
Initiate Velocity();
Evaluate Pop();
Update Fbest();
Update Pbest();
Insert nodom();
for i = 1, 2, ..., MAXCycles do
   for j = 0, 1, 2, ..., MAXParticles do
      Update Velocity();
      Update Particle();
   end for
   Evaluate Pop();
   Update Fbest();
   Update Pbest();
   Insert nodom();
   Gbestpos = rnd(0,nodomfileSize);
   Plot iteration();
end for
}
End Program
```

# Chapter 3

# IEEE 802.11 Optimization Model

In this chapter, a novel MOO model is proposed for the physical and data link layers of the 802.11 protocol. This model is solved by the PSO algorithm using a series of crossing variables in order to have common variables between layers.

## 3.1 Network Model

A model is presented of the network that represents the tasks of the physical layer and the sub-layers MAC and LLC is presented below. This model is divided in five subsections, and variables used are explained in more detail in each subsection.

## 3.2 Physical layer

For all the network layers, the physical layer is the most difficult one to understand and model, the basic reason is that the physical layer is an hybrid implementation between hardware and software. For example, in a network system, the physical layer cannot specify its channel between two connected nodes, but must instead predict it. This can cause difficulties if the channel is not ideal. Figure 3.1 shows the taxonomy of the physical layer. For example, Ekpenyong et al. [ ] modeled the throughput of the physical layer with (Eq. 3.1):

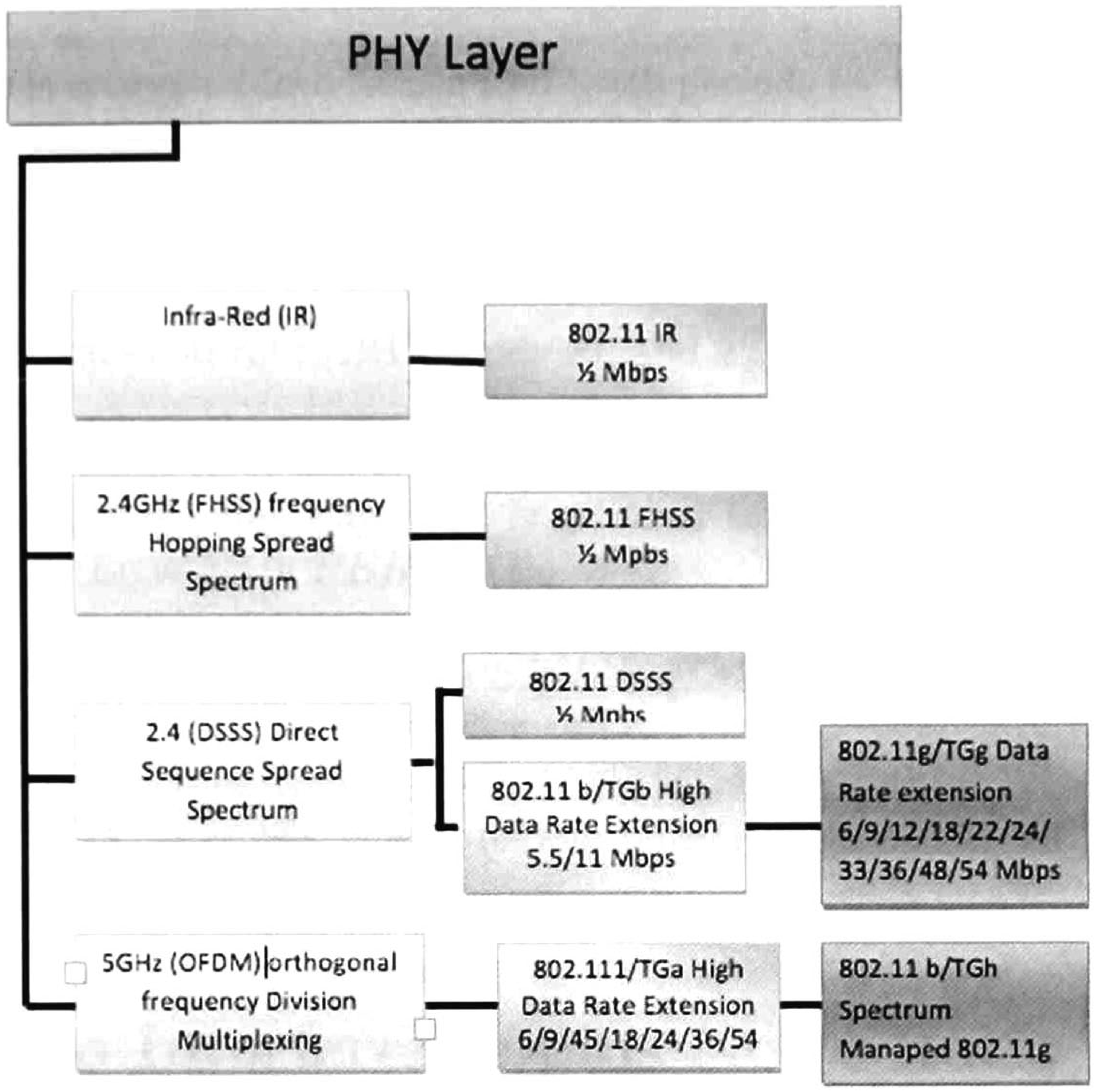


Figure 3.1: Snapshot of 802.11 PHY standard activities.

$$T = \frac{L}{L+C} bR_s PSR\left(L, b, \gamma_s\right), \tag{3.1}$$

where the variable $L$ represents the payload length in bits, $C$ is the header and DCF the overhead corresponding to rate in bits, $R$ is the data rate corresponding to PHY , $b$ is the number of bits per MQAM symbol, and $PSR_s$ is the Packet Success Rate (PSR) defined as the probability of receiving a packet correctly. Finally, $\gamma_s$ is the Signal Noise Ratio (SNR) per symbol which is represented by:

$$\gamma_s = \frac{\epsilon_s}{N_0} = \frac{P}{N_0 R_s}, \tag{3.2}$$

where $\epsilon_s$ represents the energy per symbol, $N_0$ is the one-sided noise power spectral density and $P$ is the receiving power. $C$ takes the CSMA/CA channel access time, and the header overhead into account as specified by the

IEEE 802.11 protocol.

The time delay is converted into transmited bytes periods for the purpose of optimization by the expression:

$$C = R_s * T_{ovh}, \tag{3.3}$$

where $R_s$ is the transmission rate (PHY mode $s$), and $T_{ovh}$ is the total protocol overhead.

Now, given any symbol error in the recived packet, the followed strategy is to discard the packet. In this way, PSR is given in terms of the symbol error rate and Packet Error Rate $PER$ by (Eq. 3.4):

$$PER(L, b, \gamma) = 1 - PSR(L, b, \gamma), \tag{3.4}$$

where the variable $PER$ comes from the MQAM protocol for Additive White Gaussian Noise (AWGN) channels. This previous equation, as various PHY models in IEEE 802.11a, is derived in [ ].

# 3.3 Signal to Interference plus Noise Ratio

Signal to Interference plus Noise Ratio (SINR) is important when calculating the BER. In order to use SINR in those calculations, the following equation (Eq. 3.5) is used [ ]:

$$SNIR = \frac{P_{signal}}{P_{noise} + P_{interference}}, \tag{3.5}$$

where $P_{signal}$ is the signal power, $P_{noise}$ is the noise power, and $P_{interference}$ is the interference power. In addition, SINR is expressed in dB.

SINR is commonly used in wireless communication as a way to measure the quality of wireless connections since the energy of a signal typically fades with distance. In wireless networks, this is commonly known as path loss. However, unlike wired networks, a wireless communication network has to take a lot of environmental parameters into account. Examples of these parameters are the background noise and interfering strength of other simultaneous transmission. Therefore, SINR attempts to synthesize a representation of all these environmental parameters.

## 3.4 Bit Error Rate

The BER is used in the multi-objective function because it is used as a measure of error in the different layers of the 802.11 protocol. It represents the number of bit errors divided by the total number of transferred bits during an observed time interval, and it is expressed as a percentage.

Now, it is necessary to point out that the BER changes with respect to bandwidth of the physical channel. For example, this work uses the 5.5 Mb/s bandwidth, which has (802.11 standard [ ]) the following equation (Eq. 3.6) for the BER calculation:

$$BER = \frac{8}{15}\left(14 \times Q(8 \times SINR)^{\frac{1}{2}} + Q(16 \times SINR)^{\frac{1}{2}}\right), \tag{3.6}$$

where the $Q$ function (Eq. 3.7) is defined as the area under the tail of the Gaussian probability density function with zero mean and unit variance.

$$Q(x) = \left(\frac{1}{\sqrt{2\pi}}\right)e^{\left(\frac{x^2}{2}\right)}\left(\frac{8+9x^2+x^4}{15x+10x^3+x^5}\right). \tag{3.7}$$

## 3.5 Selective-Repeat Model

IEEE 802.2 is the IEEE 802 standard and it defines the Logical Link Control (LLC) at the top of the link layer in local area networks. This LLC sublayer presents a uniform interface to the user of the data link services, which is usually the network layer. In addition, the protocol used in this sub-layer is the SR-ARQ known as selective-repeat protocol.

The selective-repeat protocol is a general strategy for handling frame transmission errors when the round-trip time for frame transmission and reception of the acknowledgment is comparable to the frame transmission time. SR-ARQ is used by the TCP transport protocol, where the transmitter groups the frames into windows so that each window contains $N$ frames. Now, when the sender transmits frames within a window, the receiver stores the frames of the current window and check for errors. After a complete window has been received or after a proper timeout period, the receiver instructs the transmitter to resend only the frames that could contain errors.

Next, it is possible to calculate the throughput for the protocol of SL-ARQ as the total number of frames transmitted by using the equation (Eq.

3.8):

$$ARQ = \frac{N}{N + N_a}. \tag{3.8}$$

where $N$ is the number of frames, $N_a$ in (Eq. 3.8) is the average number of frames sent due to all retransmissions, which is modeled by (Eq. 3.9):

$$N_a = \sum_{k=1}^{km} n_k \alpha_k. \tag{3.9}$$

Here, the variable $n_k$ in (Eq. 3.9) is the average number of frames sent at the $k_t h$ retransmissions, which is modeled by equation (Eq. 3.10):

$$n_k = \sum_{j=1}^{N} j r_{k,j}, \tag{3.10}$$

where the variable $\alpha_k$ is the probability for the source being in the $k_t h$ retransmission state. Again, this probability is modeled by equation (Eq. 3.11):

$$\alpha_k = \sum_{j=1}^{N} r_{k,j}, \tag{3.11}$$

where in the variable $r_{k,j}$ (Eq. 3.11), the index $k$ represents the states of the sender while it is retransmitting frames for the first time due to a damaged or lost frame. The index $j$ is the corresponding $j_t h$ error frame, where the $N$ is the first error frame and 1 is the last error frame.

All these equations require the probability of transmission, which is expressed as:

$$P_{i,s,d} = \prod_{s}^{i} (1 - \overline{P}_j), \tag{3.12}$$

where $P_{i,s,d}$ is the probability of transmission from node $i$ to node $d$ via node $s$.

From all the previous equation, it is possible to develop a throughput measure of the packet retransmission, which is called ARQ (Eq. 3.13):

$$ARQ = \frac{N}{N + \sum_{k=1}^{km} n_k \alpha_k}. \tag{3.13}$$

This is the final equation that it is used as part of the model of the MOO problem, due to the equation given by the throughput of the protocol ARQ.

## 3.6 CSMA/CA Protocol

The protocol at the sublayer logical link control is known as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Although, it is a popular protocol for station transmissions [ ], it has the drawback of being unable to determine if a collision had occurred while a transmission is done or not.

In [ , , , ], the authors develop several mathematical models of the sublayer MAC. These models are used in this work to generate a throughput cost function for the multi-objective optimization.

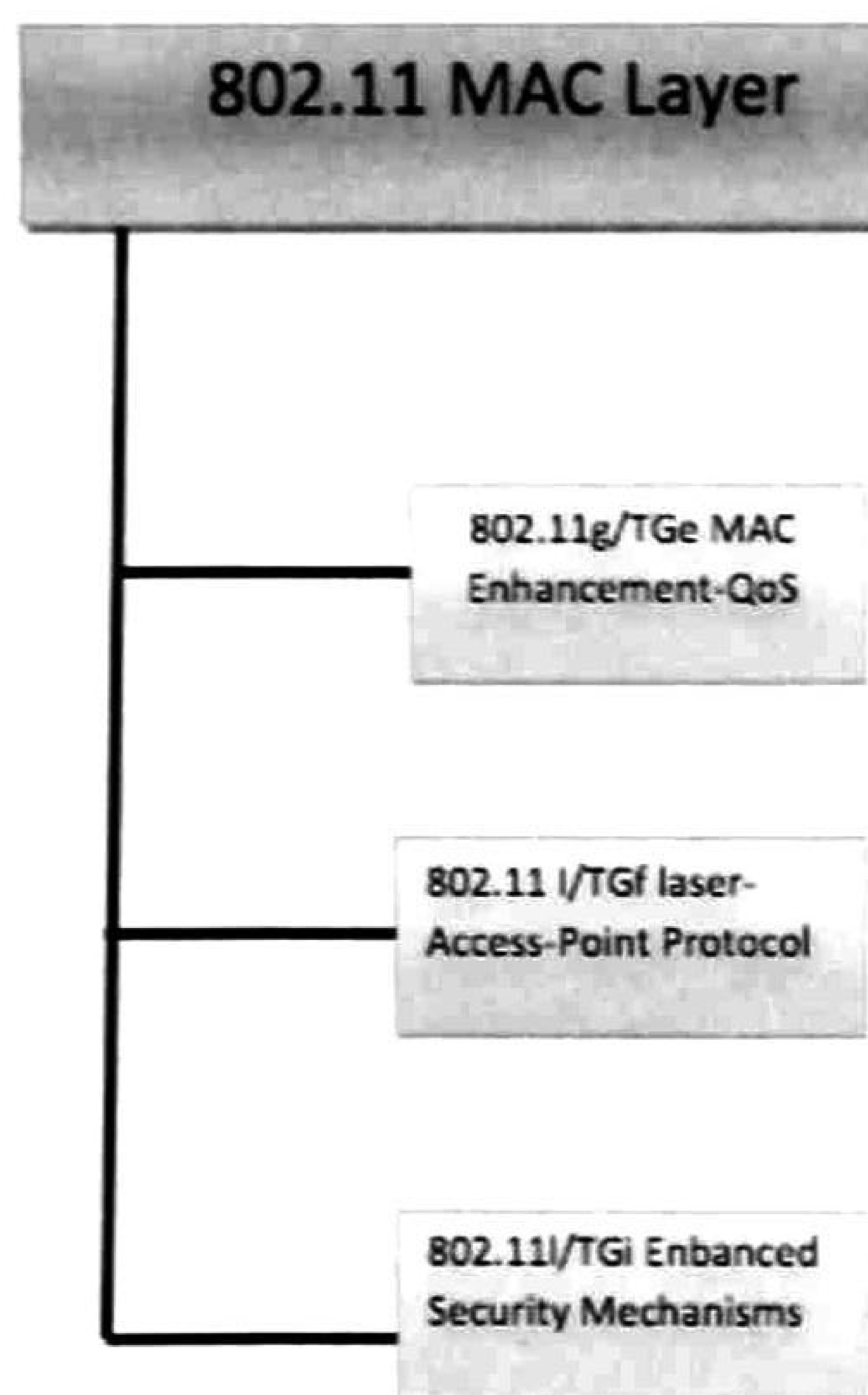


Figure 3.2: Snapshot of 802.11 MAC standard activities.

Before, the throughput equation is presented, it is necessary to explain some of the particularities of the protocol CSMA/CA. In this protocol as soon as a node receives one packet that is sent, it checks if the channel is clear. If the channel is available, then the packet is sent. If the channel is not available, the node waits for a randomly chosen period of time, and then checks again to see if the channel is available. This period of time is called the backoff factor, and is counted down by a backoff counter. If the channel is available when the backoff counter mark zero, the node transmits the packet. If the channel is not available when the backoff counter mark zero, the backoff factor is set again, and the process is repeated until it is successful.

Using the previous ideas, it is possible to define the throughput of the protocol CSMA/CA (Eq. 3.14) as:

$$\begin{aligned} \tau &= \sum_{i=1}^{m} bw_{(i,0)} + bw_{(i_0,0)e} q(1-p) = \dots \\ &= bw_{(i_0,0)e} \frac{q^2}{1-q} \left( \frac{W_0}{(1 - PSR(b, L, \gamma)) \left(1 - (1-q)^{w0}\right)} \right) \end{aligned} \tag{3.14}$$

where $bw_{(i,0)}$ and $bw_{(i_0,0)e}$ represent the waiting periods of a node in a backoff state with waiting frames and not waiting frames respectively. Finally, the collision probability is $p$, the steady state probability is $b$, and $q$ measures the relationship between the per-station.

Another useful metric is the MAC delay. For example, consider the situation that arises immediately after a node $t$ completes a transmission. Then, the node begins a post backoff choosing a backoff $k$, and a packet arrives after $j$ states. Thus, the average time between the packet arrival at the MAC sublayer and the completed transmission is:

$$\Delta_t = \sum_{k=0}^{W_0} \frac{1}{k} \sum_{j=0}^{\infty} q(1-q)^j \Delta_{tjk}, \tag{3.15}$$

where $W_0$ is defined as a two dimensional Markov chain for the CSMA/CA process. Thus, $W(t)$ is defined by (Eq. 3.16):

$$W(t) = (sw(t), bw(t))\, or (0, bw(t)), \tag{3.16}$$

if the node is in a backoff period regardless of the number of waiting frames or not. Next. $\Delta_{ijk}$ is defined in (Eq. 3.17):

$$\Delta_{ijk} = (k-j)\, E_s + (1-p)\, T + p\, (T_{cl} + K_1)\,, \tag{3.17}$$

where $E_s$ is the expected state length if source $l$ is silent. In addition, $T_0$ is the expected length of a collision involving source $l$, and $K_0$ is the expected time for $l$ to transmit a frame beginning with a stage 0-backoff.

# Chapter 4

# IEEE 802.11 Mutil-Objective Solution

A Multi-Objective Optimization (MOO) problem consists of several possible cost functions. In the proposed model, the equations defined in the previous sections are used to derive the MOO cost function.

The first cost function corresponds to the throughput of the physical layer. This cost function is defined as follows:

$$f_1(R, L, b) = \frac{L}{L+C} bRPSR(b, L). \tag{4.1}$$

Because this equation measures the throughput at the physical layer, this objective function will be maximized.

For the second cost function, the ARQ throughput is used to represent the data link layer/sub-layer LLC (Eq. 4.7).

$$f_2(b, L, \gamma) = \frac{bw_{(i_0,0)e}q^2}{1-q} \times \left( \frac{W_0}{(1 - PSR(b, L, \gamma))\left(1 - (1-q)^{w0}\right)} \right) \tag{4.2}$$

In (Eq. 4.2), one of the major benefits is that the message confirmation has been reduced for better network performance. Thus, it is necessary to minimize this objective function for the MOO problem.

The last cost function is the CSMA/CA throughput (Eq. 4.3):

$$f_3(n_k, \alpha_k) = \frac{N}{N + \sum_{k=1}^{km} n_k \alpha_k} \tag{4.3}$$

In this last equation (Eq. 4.3), it is necessary to maximize the throughput of the logical link control sublayer, since it models the throughput of the logical link control.

At this point, the model is an abstraction of the physical layer, media access control and logical link control sub-layers, but it is not yet ready for the PSO stage. To achieve this, it is necessary to do some adjustments before it is ready for the PSO algorithm, and this is done using the CLD approach (Seq. 2.4).

## 4.1 The Crossing Variables

In order to use the PSO algorithm for the proposed MOO, it is necessary to modify and introduce certain variables at the original formulations. These are going to be called Crossing Variables (CV) in the different objective functions, and these CV variables are a result of using the concept of PSR.

In order to derive the CV variables, at the proposed MOO, the probability of receiving a packet correctly is given by (Eq. 4.4):

$$PSR(L, b, \gamma) = 1 - PER(L, b, \gamma). \tag{4.4}$$

Then, it is possible to modify the $f_1$ equation by taking into account that this equation can be seen as the expression of the physical layer performance the previos equation 4.1 is take:

$$f_1(L, b, R, \gamma) = \frac{L}{L + C} b \; R \; PSR(L, b, \gamma).$$

Next, it is necessary to modify $f_3$ in order to have the correct CV's. First, (Eq. 3.8) includes the PSR, which has the following property (Eq. 4.5):

$$1 = PSR(L, b, \gamma) + PER(L, b, \gamma). \tag{4.5}$$

This indicates that the total package rate is the total proportion of the successful packets transmitted plus unsuccessful error packet (No matter which ones). Then, when the original equation (Eq. 3.13) is combined with the (Eq. 4.5) produces (Eq. 4.6):

$$f_3(b, \gamma, n_k, \alpha_k) = \frac{N \left(PSR\left(L, b, \gamma\right) + PER\left(L, b, \gamma\right)\right)}{N + \sum_{k=1}^{km} n_k \alpha_k} \tag{4.6}$$

A side effect of using the $PSR$ function is that (Eq. 4.7) can be modified in the following way:

$$f_2(b, L, \gamma) = \left(\frac{bw_{(i_0,0)e}q^2}{1-q}\right)\left(\frac{W_0}{(1-PSR(b,L,\gamma))\left(1-(1-q)^{w0}\right)}\right) \tag{4.7}$$

The key in the crossing of variables is the packet success rate, which can be manipulated to work at layer 1 and layer 2. This allows the sharing variables in the model, thus the proposed MOO model can work with the PSO algorithm.

Finally, but not less important, the metric of the bandwidth is a measure of data shared between all the equation representing the layers under consideration. In the proposed model, a bandwidth of 5.5 Mb/s is used because we believe it helps us in all our simulations use the same bandwidth, so this does not have to be a factor that alters our results.

## 4.2 Muti-Objective Model

The final proposed model used in this work after making the necessary adjustments for the physical layer (Eq. 4.8) is:

$$f_1(R, b, L, \gamma) = \frac{L}{L+C} b\ R\ PSR(L, b, \gamma)\,. \tag{4.8}$$

The logical link control sublayer the equation is present in (Eq. 4.9):

$$f_2(b, L, \gamma) = \frac{bw_{(i_0,0)e}q^2}{1-q} \times \left(\frac{W_0}{(1-PSR(b,L,\gamma))\left(1-(1-q)^{w0}\right)}\right) \tag{4.9}$$

The last cost function is the CSMA/CA throughput (Eq. 4.10):

$$f_3(b, L, \gamma, n_k, \alpha_k) = \left(\frac{N\left(PSR(L,b,\gamma) + PER(L,b,\gamma)\right)}{N + \sum_{k=1}^{km} n_k \alpha_k}\right) \tag{4.10}$$

## 4.3 Constraints definition

In Wireless Networks, the constraints play an important role in the success of a given design. Hard constraints mean higher design effort, this deliver the need for automated tools to guide the designer into the critical design decisions. In most of the cases, the constraints are the software and the maximum available area for hardware. In this case, because it is a simulation, the constrains are based on the restrictions on the available Network.

### Constraints modeling

Now, it is necessary to formulate the constraints that allow maximizing the throughput of the network. Actually, the maximization of the throughput is equal to:

$$(\max f_1, \min f_2, \max f_3) \tag{4.11}$$

subject to the following constraints based on the protocol 802.11xx:

1. All the cost functions should have a lower limit for physical reasons implicit in each layer transmission. Thus, for the proposed multi-objective model, each function has the following lower limits:

$$f_1 \geq 0, f_2 \geq 0, f_3 \geq 0. \tag{4.12}$$

2. The channel is shared among $N$ stations according the CSMA protocol.
3. The frame arrival probability per time step is $a$.
4. All transmitted frames have equal lengths.
5. A $p - persistent$ CSMA/CA is assumed.
6. The flow conservation equation is satisfied over each frame of $N$ time slots.
7. The maximum rate that can be sent over each link is $(1 + KlogSINR)$.

Using these constraints, it is possible to use the PSO algorithm.

## 4.4 PSO Algorithm

This section explains the algorithm used to optimize the proposed multi-objective model in detail. This algorithm is based on the PSO [ ] by James Kennedy and Russell Eberhat (Sec. 2.7).

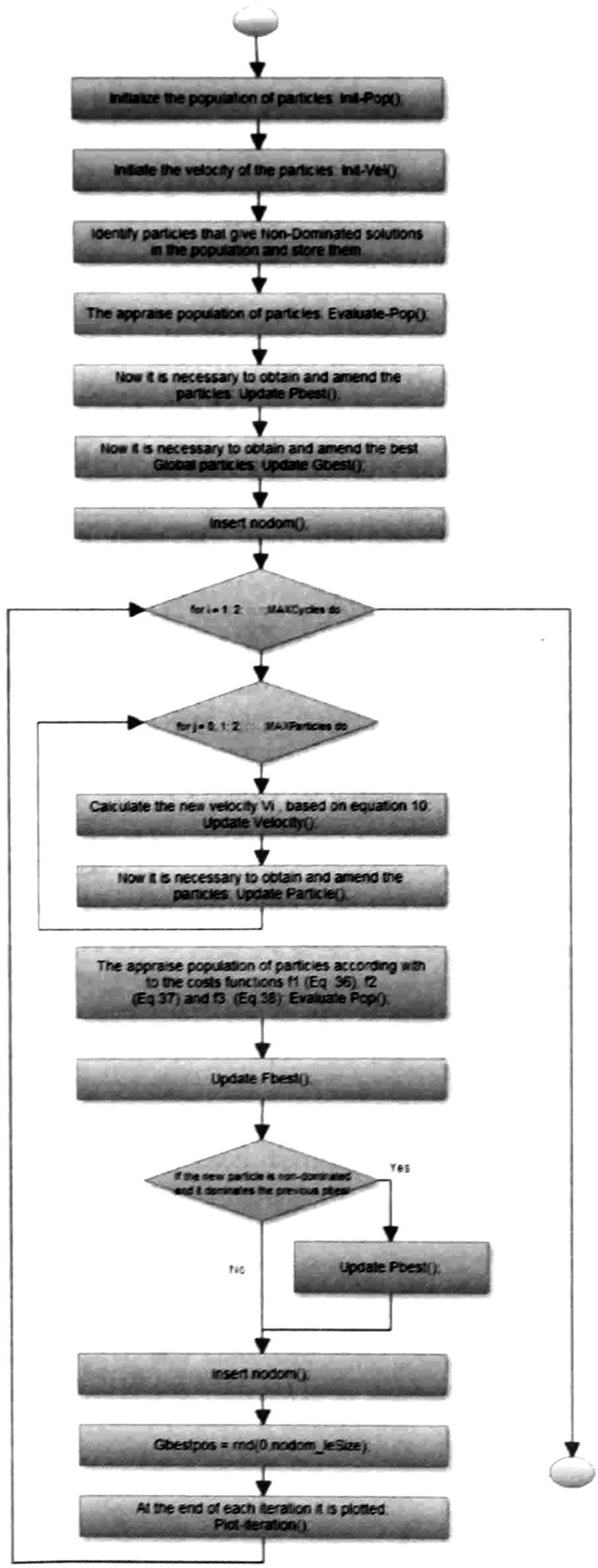


Figure 4.1: The used PSO Algorithm.

The algorithm works in the following way: first, an initial population is selected with an initial velocity, next, these particles are evaluated to obtain the non-dominated ones, and finally, this is repeated at each cycle of the PSO. A more detailed explanation can be seen in (Algorithm 4.4). as shown below

**Algorithm 2** Particle Swarm Optimization Algorithm

PSO {
Initialize the population of particles: Init-Pop();
Initiate the velocity of the particles: Init-Vel();
Identify particles that give Non-Dominated solutions in the population and store them.
The appraise population of particles: Evaluate-Pop();
Now it is necessary to obtain and amend the particles: Update Pbest();
Now it is necessary to obtain and amend the best Global particles: Update Gbest();
Insert nodom();
**for** $i = 1, 2, \ldots$,MAXCycles **do**
  **for** $j = 0, 1, 2, \ldots$,MAXParticles **do**
    Calculate the new velocity $V_i$ based on equation 2.10: Update Velocity();
    Now it is necessary to obtain and amend the particles: Update Particle();
  **end for**
  The appraise population of particles according with to the costs functions $f_1$ (Eq. 4.8), $f_2$ (Eq.4.9) and $f_3$: (Eq.4.10): Evaluate Pop();
  Update Fbest();
  If the new particle is non-dominated and it dominates the previous pbest then updates pbest Update Pbest();
  Insert nodom();
  Gbestpos = rnd(0,nodomfileSize);
  At the end of each iteration it is plotted: Plot-iteration();
**end for**
}
End Program

It is easy to see that the PSO basically is a search algorithm of non-dominated particles. Thus, it is necessary to look through the set of non-dominated particles for a possible solution. This solution should be compared to a solution provided by the implementation of the original protocol in the NS2.

### Convergence of PSO

In order to prove the convergence of the PSO [ ], it is possible to use the idea of the convergence of a sequence $\{a_i\}_{i=1}^{\infty}$, where $a_i$ is a sequence of particles being generated by the PSO. Then, it is possible to say that a sequence converges, if

$$\lim_{i \to \infty} a_i = C, \tag{4.13}$$

where $C$ is a constant particle. However, to prove convergency using this method tends to be a difficult task, and it is easier to define convergence through the use of the variance of the PSO's fitness function. This variance is defined as follows:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{f_i - f_{avg}}{f} \right)^2 \tag{4.14}$$

where $N$ is the total number of particles, $f_i$ is the fitness of the $i$-th particle in the sequence, $f_{avg}$ represents the present average fitness of the entire swarm of particles, and $f$ is the normalized calibration factor to normalize $\sigma^2$ Now, to obtain the value of $f$, it is possible to use the following equation (Eq. 4.15):

$$f = max \left\{1, max \left\{|f_i - f_{avg}|\right\}\right\}. \tag{4.15}$$

with $i \in [1, N]$. The definition of $\sigma^2$ allows to define a convergence of the particles in the PSO swarm. For example, a smaller $\sigma^2$ is a signal of a better convergence. Then, the particles convergence in the swarm can be defined as:

$$\lim_{t \to +\infty} X(t) = \alpha \times pBest + (1 - \alpha) \times gBest, \tag{4.16}$$

where $X(t)$ denotes the position of a particle at time $t$, $\alpha = \frac{\psi_1}{\psi_1 + \psi_2}$, where $\psi_1$ and $\psi_2$ are the acceleration coefficients in (Eq. 2.10), $pBest$ denotes the individual extreme of this particle while $gBest$ is the global extreme of the particle swarm. According to theorem presented in [ ], the variance of the populations fitness, $\sigma^2$ (Eq. 4.14) is zero. This is how it is possible to prove that the PSO algorithm converges to the optimal solution.

# Chapter 5

# Experimental Results

For the Multi-Objective Optimization (Sec. 2.6), the density of the non-dominated solutions (Def. 2) is directly associated to the population size. Therefore, a large population size was required in order make an effective search of the solution space. For this reason, the population size was selected to be of 1,000 particles for the physical and CSMA functions and 1,500 particles for the ARQ function 5.1. This difference, between the populations at each cost function, was because each of the solution spaces at each function had different characteristics, making it necessary to have a minimal number of particles to obtain a sufficiently dense Pareto frontier. Thus, through trial and error, it was possible to derive the previously mentioned population sizes.

Next, the maximum number of iterations for all cases was selected to be 10,000 iterations, because at this point, the results of the algorithm started to change the condition of dominance between particles. This is explained because the PSO, an evolutionary algorithm, can overshoot the optimal solution [ ].

Finally, in order to obtain a correct statistical result, fifteen independent runs have been performed for each case. After that, an average was taken over the particle results.

In this section we want to model the throughput of the first two layers of the OSI model.

The main objectives considered are:

1. Obtain a prediction model in wireless networks.

2. Design a prototype that implements the model.

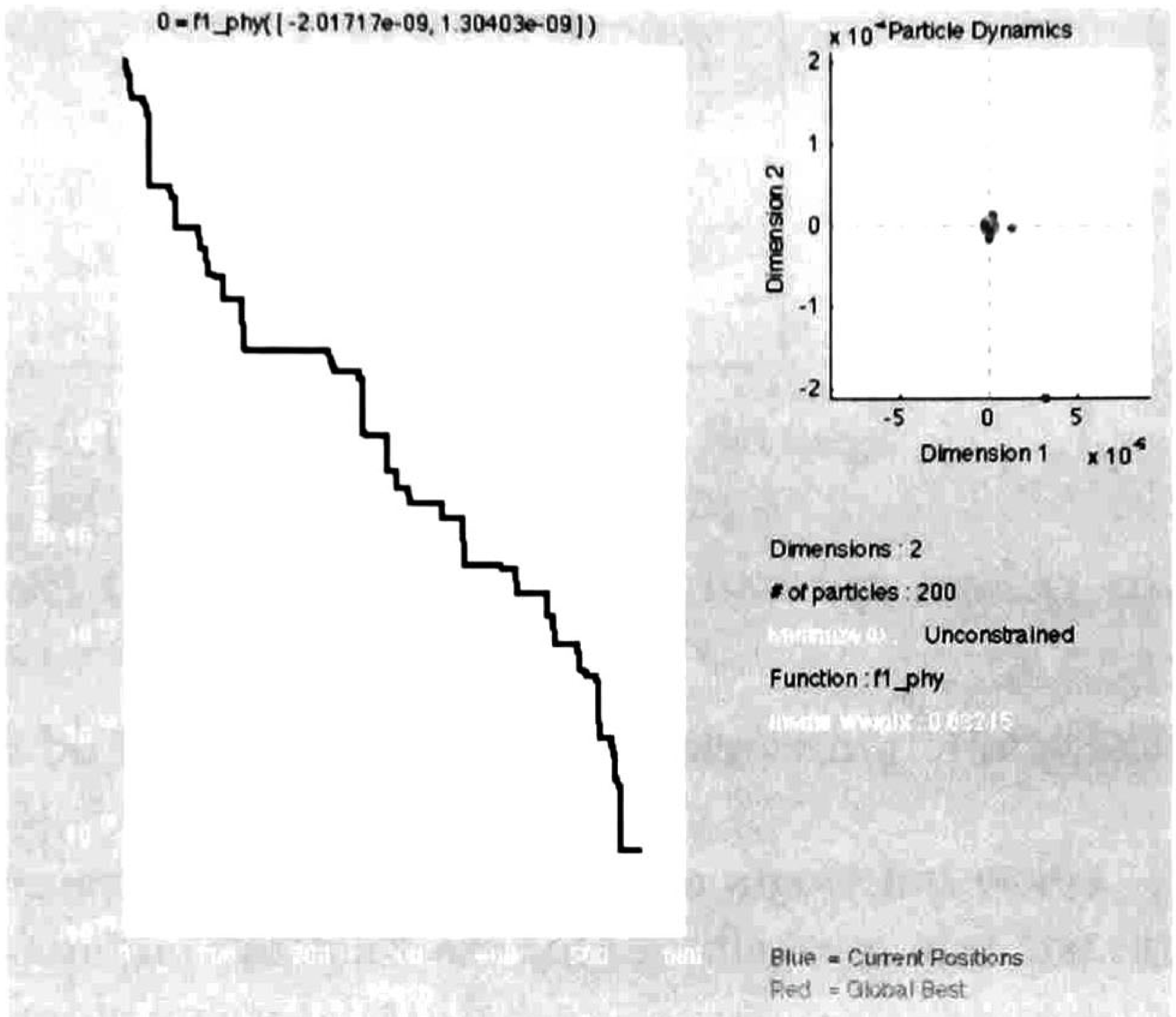


Figure 5.1: Framework.

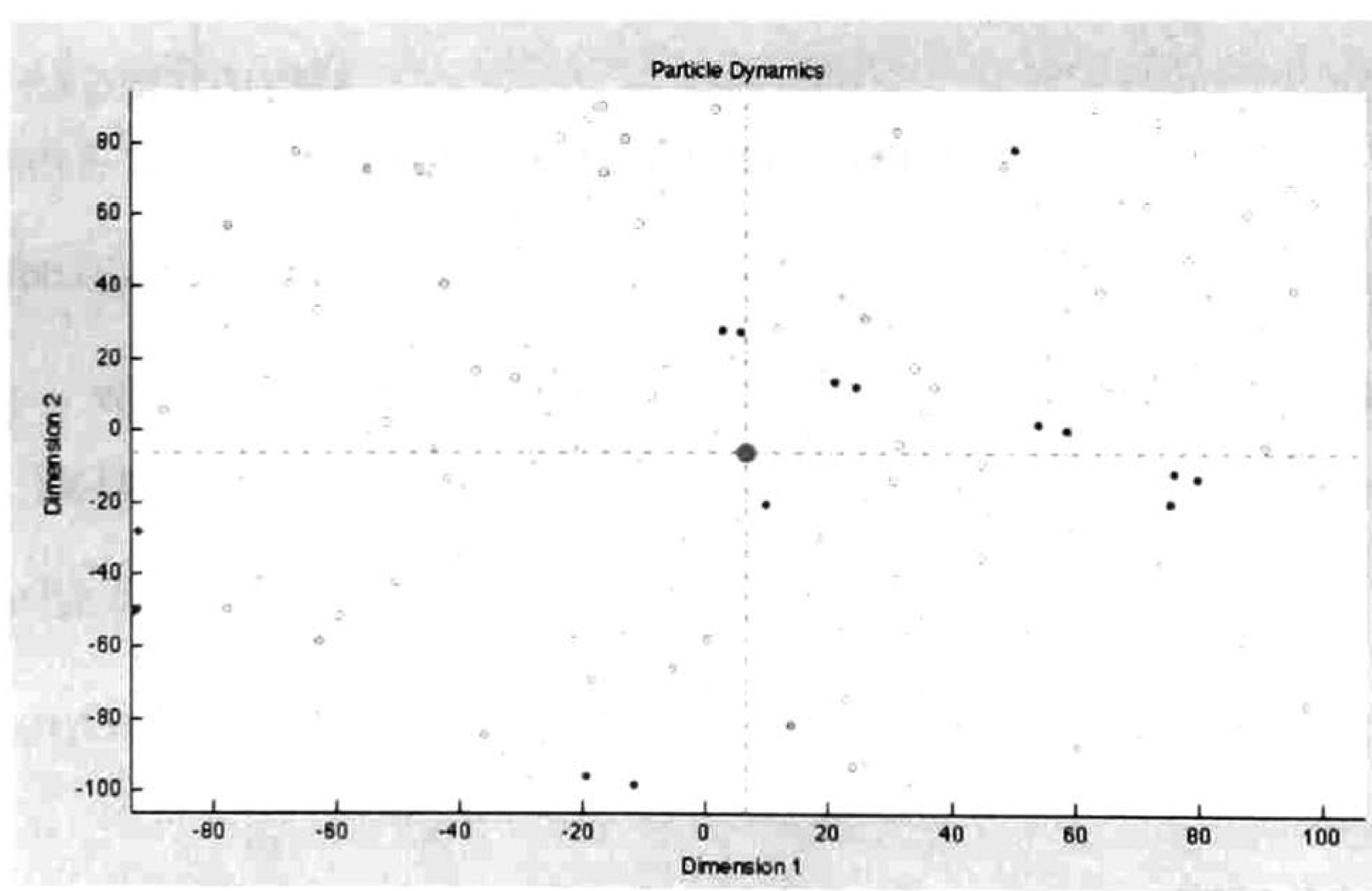


Figure 5.2: Initial sampling of the particles.

| Test Cases | Variables | Values |
|---|---|---|
| 1000 | $L$ | 180 |
| 1000 | $\gamma$ | 2.5 |
| 1000 | $b$ | 40 |
| 1500 | $L$ | 150 |
| 1500 | $\gamma$ | 2.5 |
| 1500 | $b$ | 60 |

Table 5.1: PSO Control Variable Settings

3. an use that model that allows comparisons between models and real data.

The next step will be to use an optimization algorithm: Particle swarm optimization (PSO)[ ].

In Fig. 5.2 it is presented how the optimization algorithm works. At the begging of the algorithm the particles are totally disperse and the margins are between all the sample space (-100,100).

Fig. 5.3 show results up to 200 and the particles are plot in green, it is shown the Pareto frontier and in blue all the non-dominated particles, in this case due that the particles are randomly initialize, the sample space in both the x-axis and the y-axis has negative values but this will be corrected slowly, until all particles are positive only, this is due to the nature of the variables used in work, because it is known that there are no negative values for any reason in the experiments.

The PSO parameters used at each experiment were:

1. The initial constants are: $c_1 = 1$, $c_2 = 2$.

2. $V_{max} = 0.1(UL)$ where $U$ and $L$ are the upper and lower boundaries for the decision variables.

3. The inertia weight $w = 0.1$.

4. $\chi = 0.63$, a potential solution is considered feasible when its $\Phi < \epsilon = 1.0\ E05$.

The algorithm inputs will be the specified cost functions (after editing and put in appropriate units) from the physical layer, data link layer sublayer MAC and data link layer sublayer LLC. The simulation results and the performance of the algorithm are presented next.

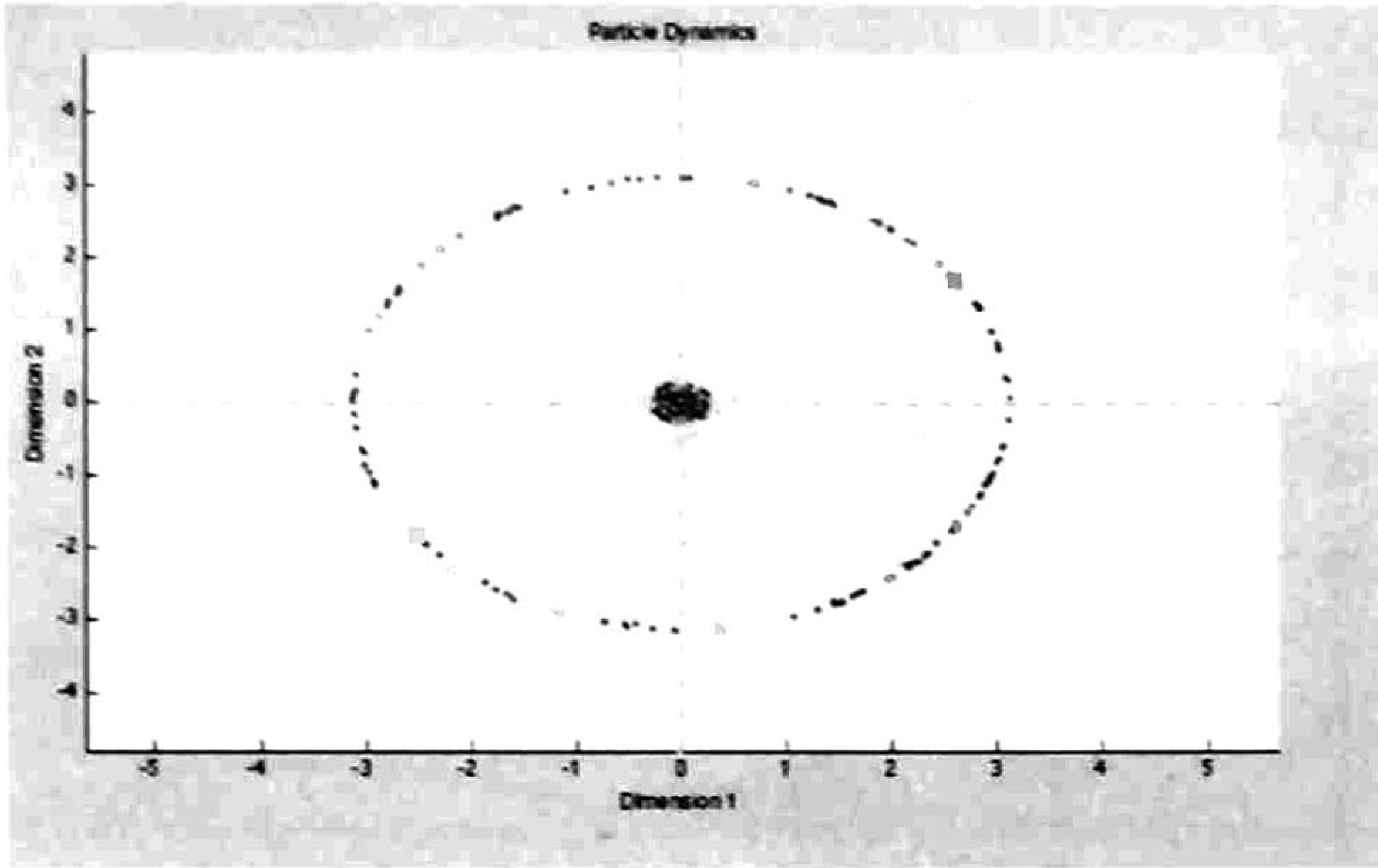


Figure 5.3: Results after few iterations.

Now the following figures Fig. 5.4 - Fig. 5.6 present the simulation results to illustrate the simulated Pareto-Optimal frontier and the performance of the algorithm, in this figures it is shown that in all the cases it is obtained the minimization expected, due to in the optimization problem the packet error rate is the variable that allows to minimize the cost funtions.

For example in Fig. 5.4 shows the optimization for the physical layer $f_1$(Eq. 1.8) and to try to maximizing the packet success rate. It is possible to see that particles non-domi are get into the Pareto frontier according with [Def. 3].

As early experiments changes, were made only at the physical layer, in this case the improvement was not significant, in fact can be said that it was minimal, as shown in the results of the table 5.2, at first view, according to the results obtained by the PSO optimization algorithm is able to obtain an improvement in the packets sent from an average of 1%.

| Test Cases | PSO | IEEE 802.11 |
|---|---|---|
| 1 | 948 | 933 |
| 2 | 881 | 870 |
| 3 | 791 | 777 |
| 4 | 649 | 640 |

Table 5.2: Results only with the physical layer

In Fig. 5.5, it is possible to see the results by introducing the formula

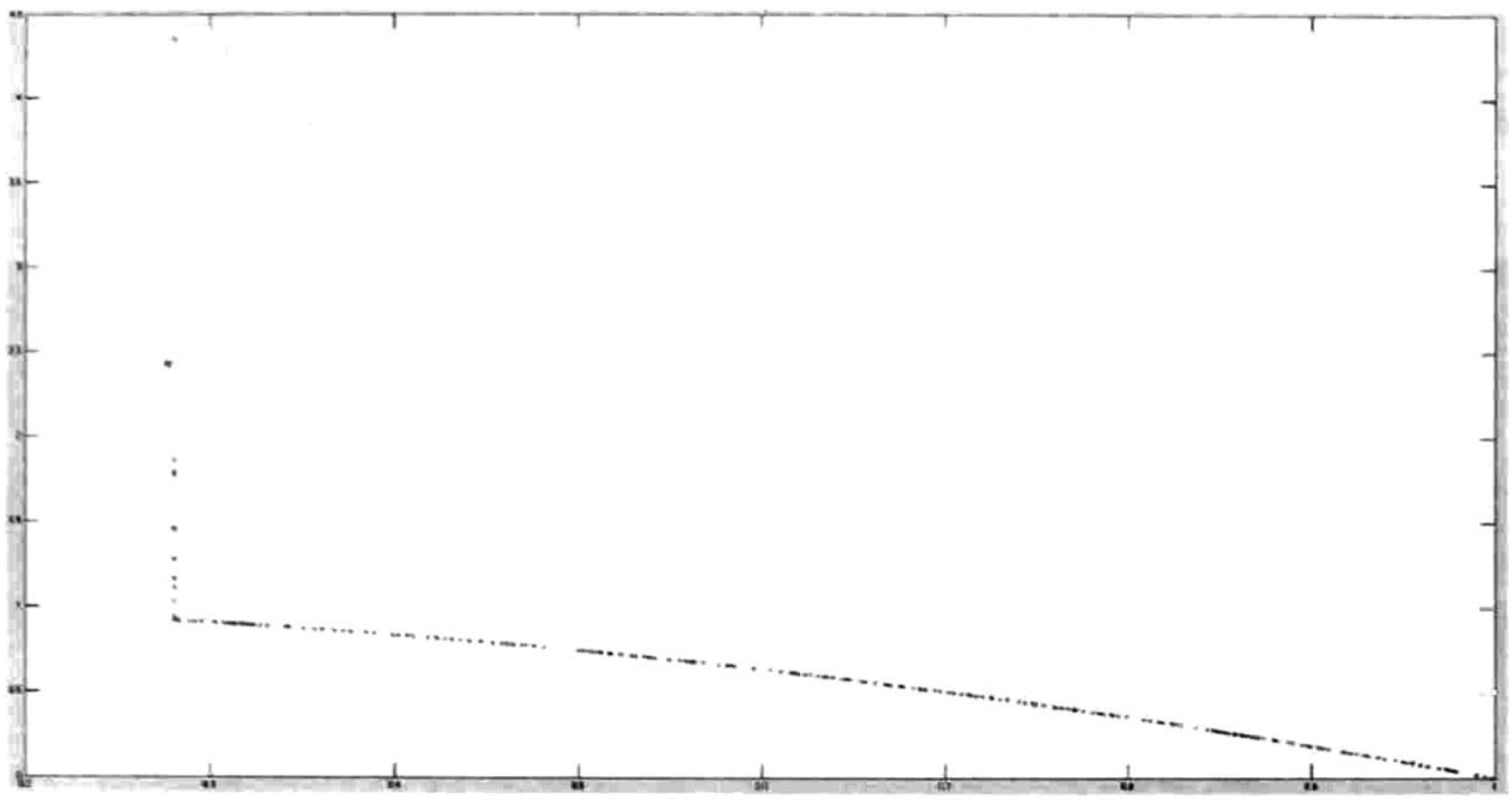

Figure 5.4: Results of the PSO algorithm are shown the Pareto frontier of the physical layer.

of the physical layer $f_1$ (Eq. 4.8) and CSMA (4.9 $f_2$). In this case, $f_1$ was maximized and $f_2$ was minimized.

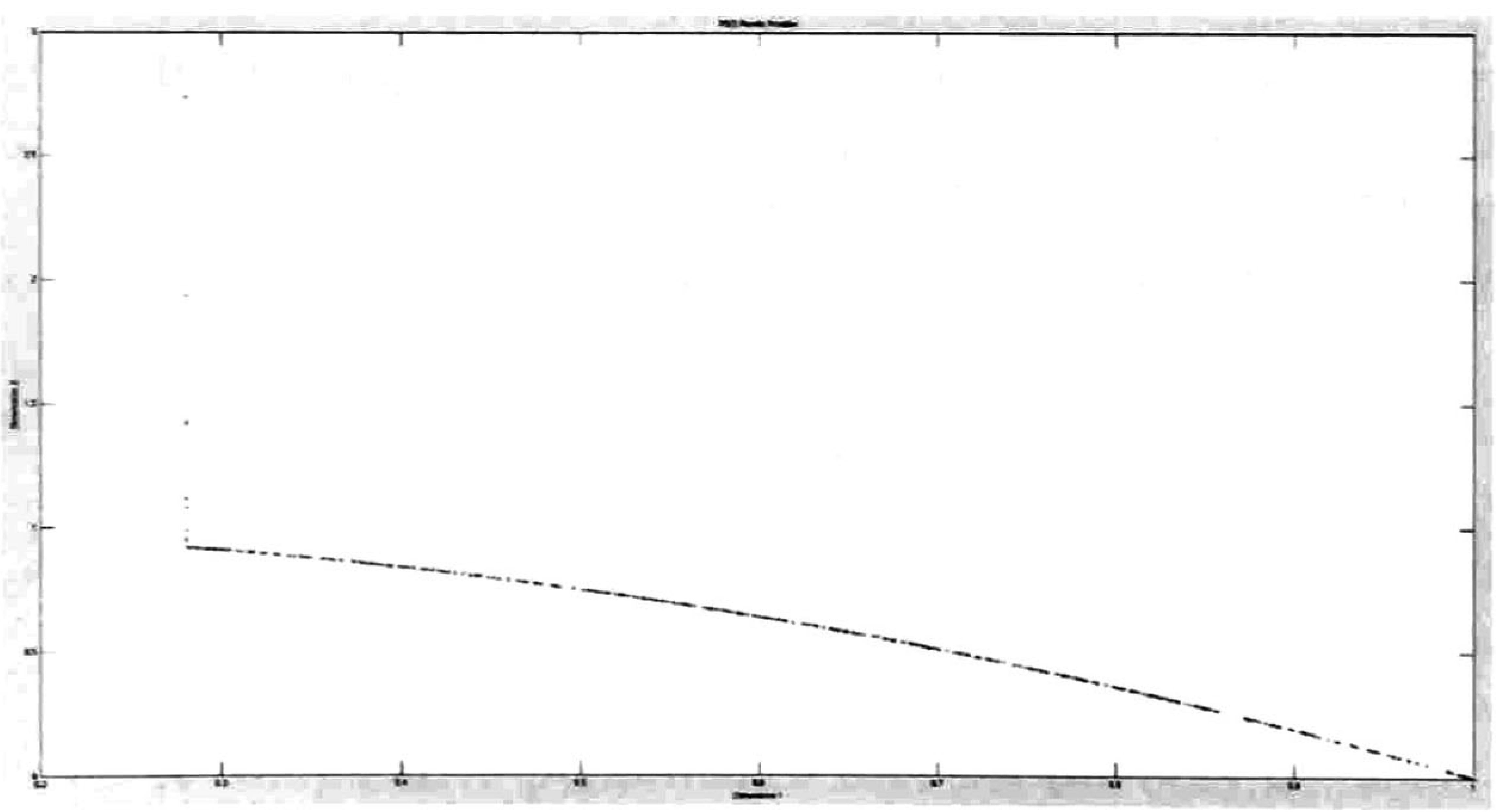

Figure 5.5: Results of the PSO algorithm are shown in the Pareto frontier of the protocol CSMA.

(Fig. ??) shows the results of the three cost funtions. This proves that the proposed optimization model has no problems and was able to obtain the Pareto frontier and that the results must now be validated in network simulator. In each of the three figures (Fig. 5.4), (Fig. 5.6) and (Fig. 5.5) it is proven that the algorithm works and the particles that optimizes the formulas are obtained.

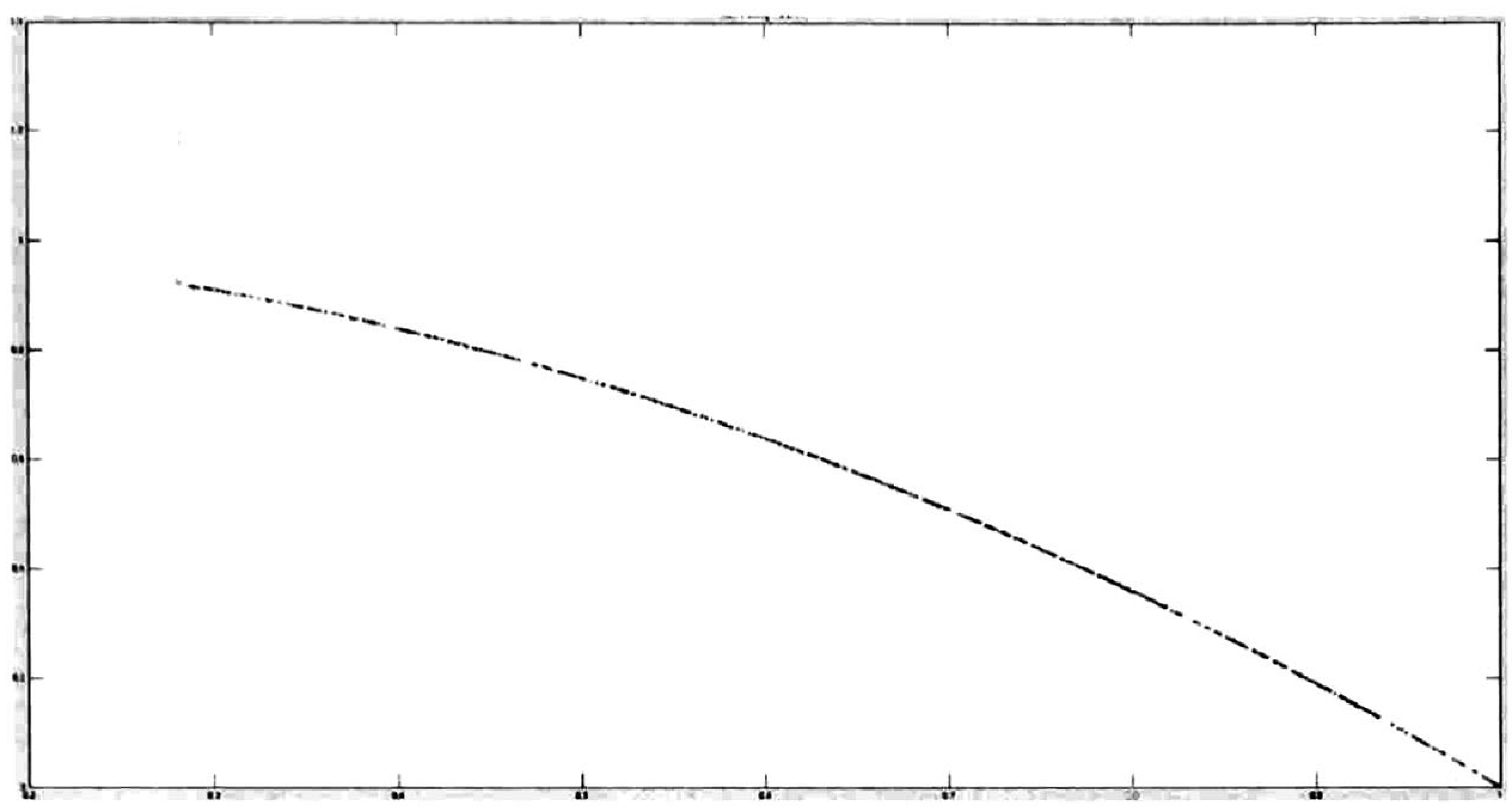

Figure 5.6: Results of the PSO algorithm are shown in the Pareto frontier of the protocol ARQ.

In the (Fig. 5.1) it is shown an example of the framework employed in this work, we perform each iteration and we stop the algorithm until we don't have a new non-dominated particle, or we stop in the 500 iteration to avoid that a particle is mixing between dominated and non-dominated.

## 5.1 NS-2 Validation

In the results are shown that it is obtained the Pareto frontier in each of the multi-objective formulas, which tells us that there is space for improvements in the design pattern of 802.11.

A comparative performance between the original model and introduced modifications in the logical link layer shows an improvement of 3 to 5% are obtained, in (Fig. 5.9 and 5.10), that shows that the throughput in the sample simulations is increased.

T

In the simulation in NS-2, two case studies were performed, the first case is made with 100 nodes (Fig. 5.8) representing a normal workplace in the real life. In the second simulation, is simulated with 1000 nodes, which was

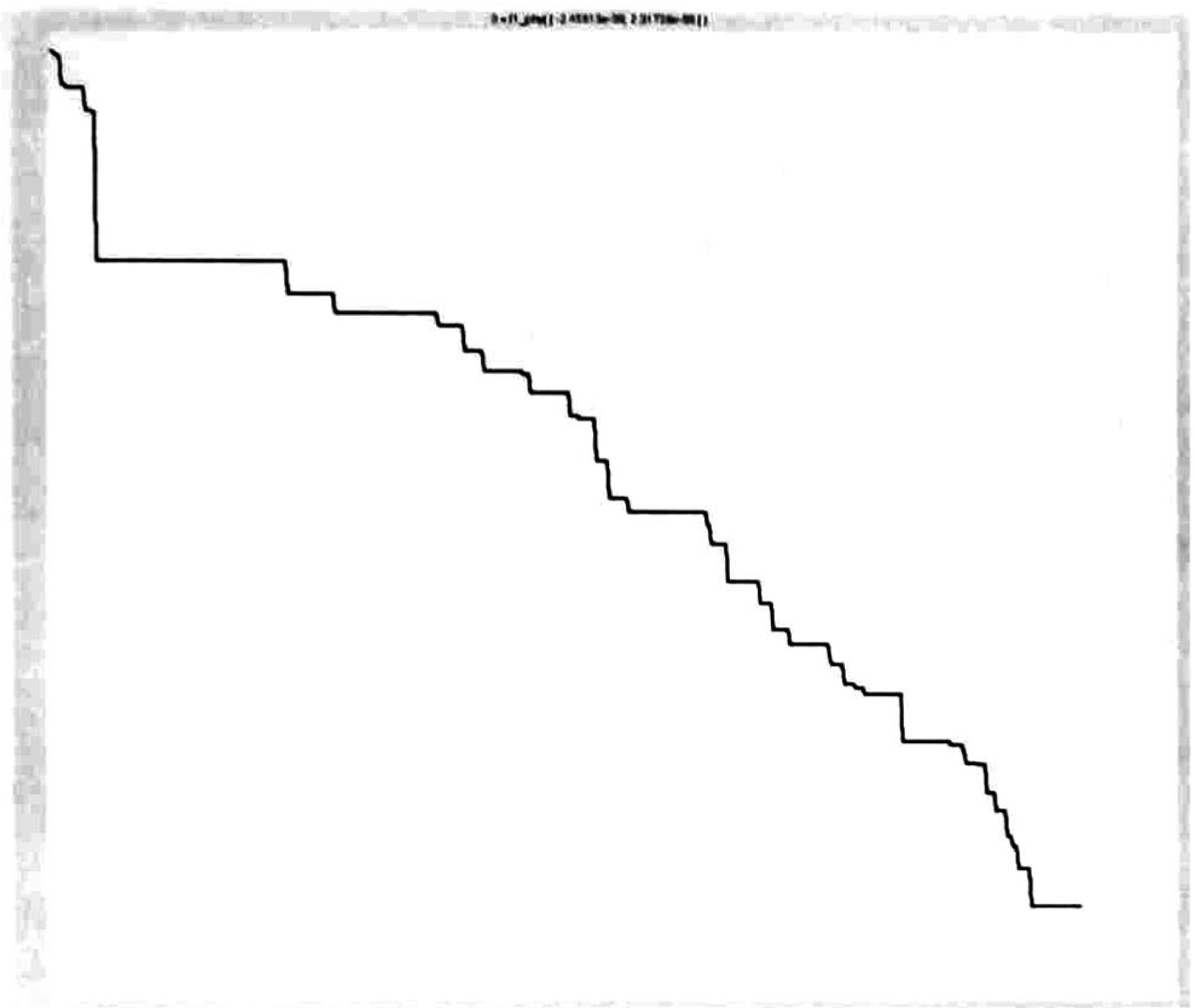

Figure 5.7: The evolution of the error.

made with the objective to bring the model to its "limits" and the results of this study are presented below.

In the simulation in NS-2, two case studies were performed, in which in the first case is made with 100 nodes representing a normal workplace in the real life, and a second simulation, in where the result is simulated with 1000 nodes, which was made with the objective to obtain the results under stress network conditions, and in which the results are presented below. In all the cases the results are equivalents, this means that there not In all cases the results obtained are equivalent, meaning that no modifications made in the scenarios simulation, such as changing the mobility of nodes, the number of nodes near the timeout and do not affect our results therefore is important to focus on the variables of the PSR.

| Test Cases | PSO | IEEE 802.11 |
|---|---|---|
| 1 | 1022 | 933 |
| 2 | 924 | 870 |
| 3 | 813 | 777 |
| 4 | 701 | 640 |

Table 5.3: Comparative between Standard and PSO with 1000 nodes

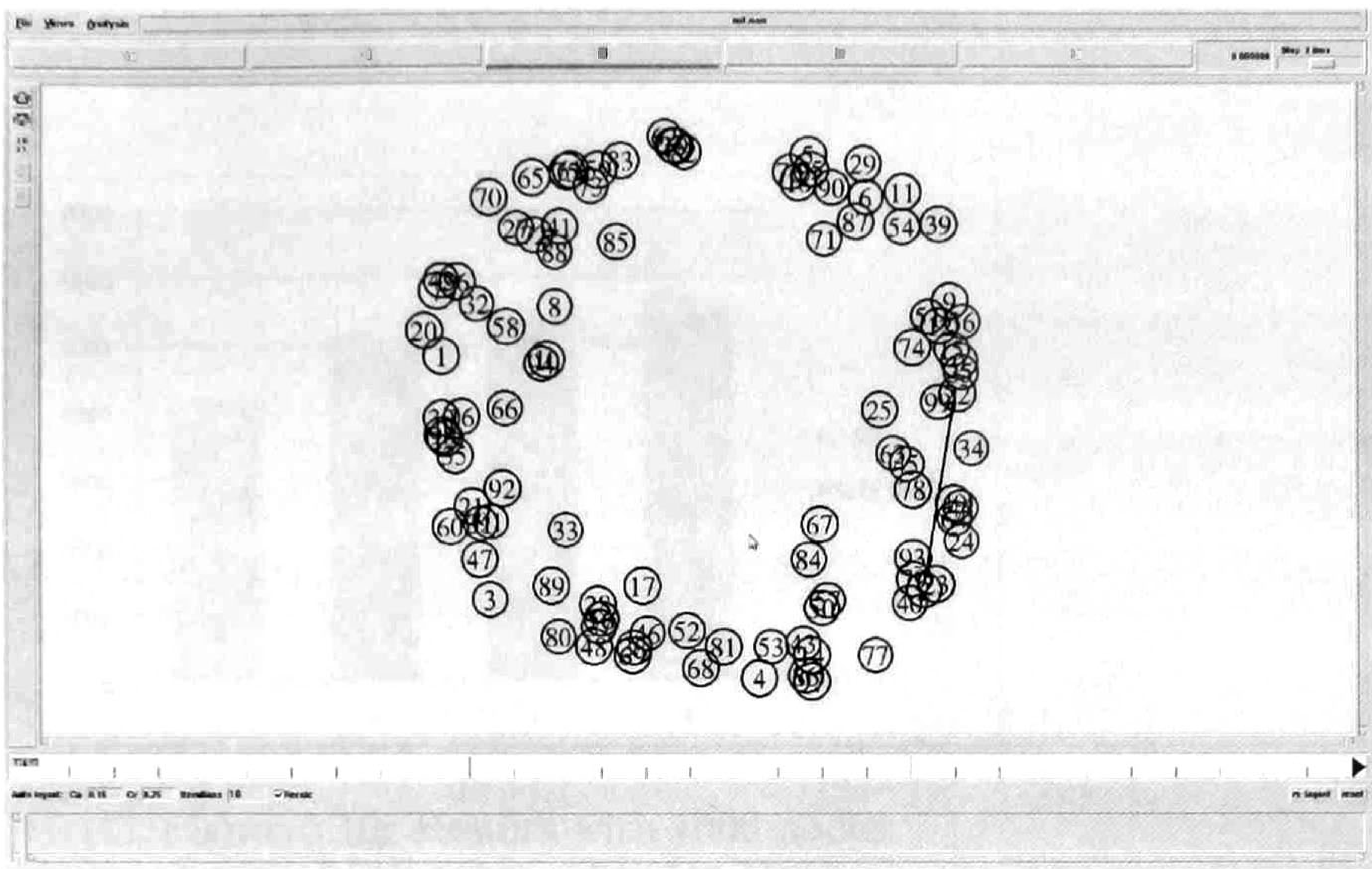

Figure 5.8: Simulation in NS2 with 100 nodes

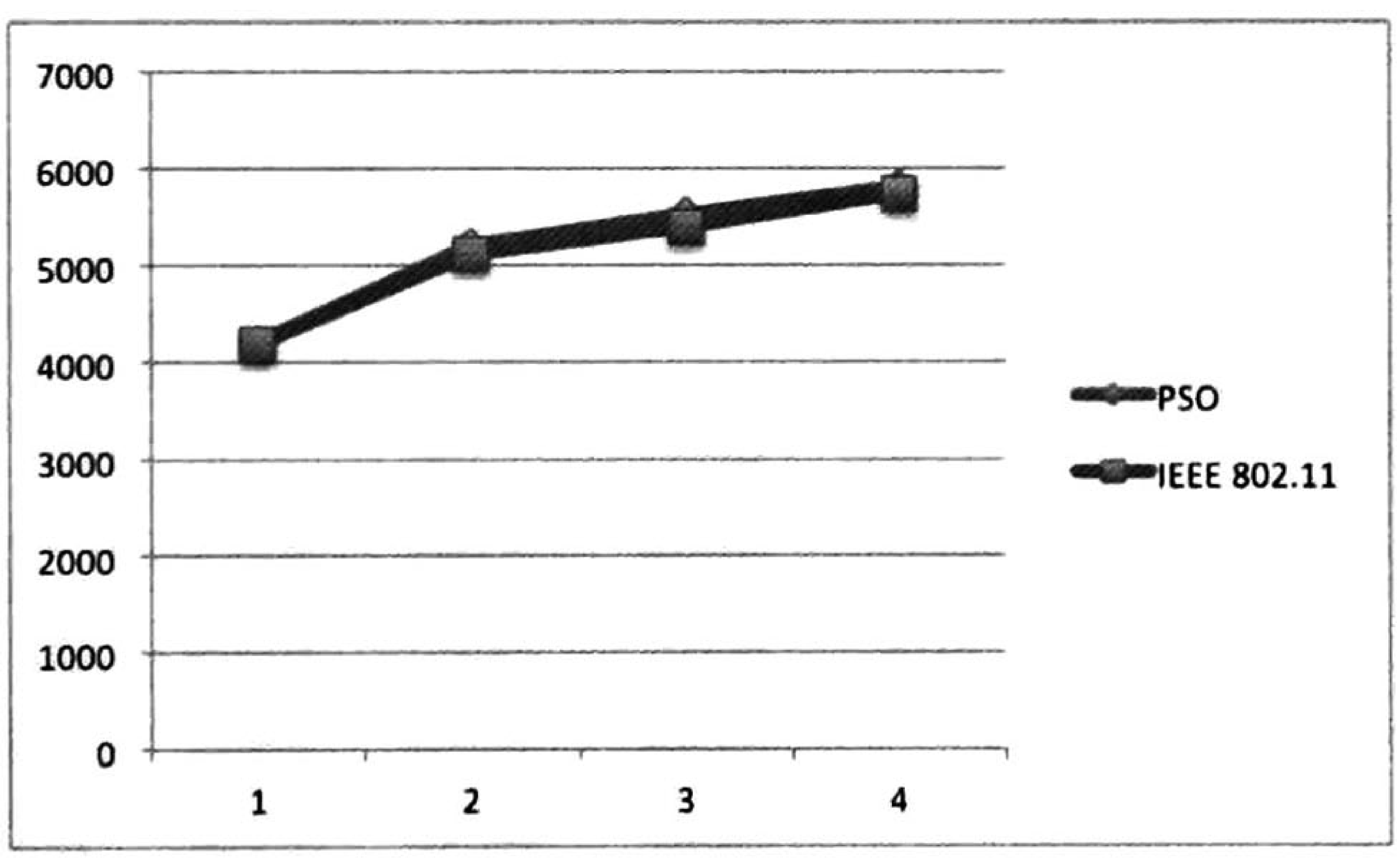


Figure 5.9: Results with 1000 nodes.

The figures (Figure 5.11) and (Figure 5.12) refer to the measurement of the throughput obtained for the scenario where nodes 1000 are simulated to achieve an improvement of 2-4% depending on the case. As in the previous figures, it can be observed that the throughput increases when bandwidth increases.

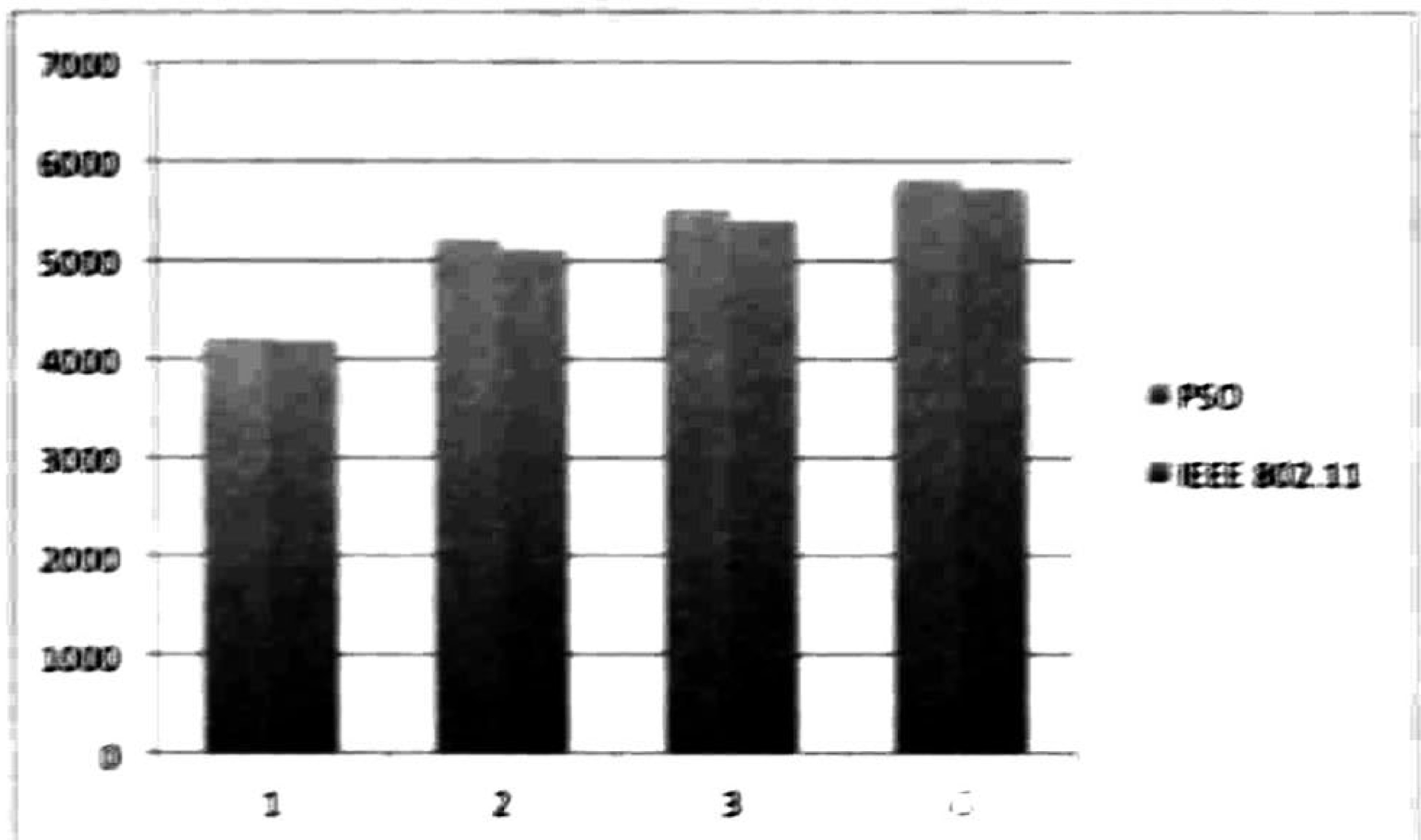


Figure 5.10: Results with 1000 nodes.

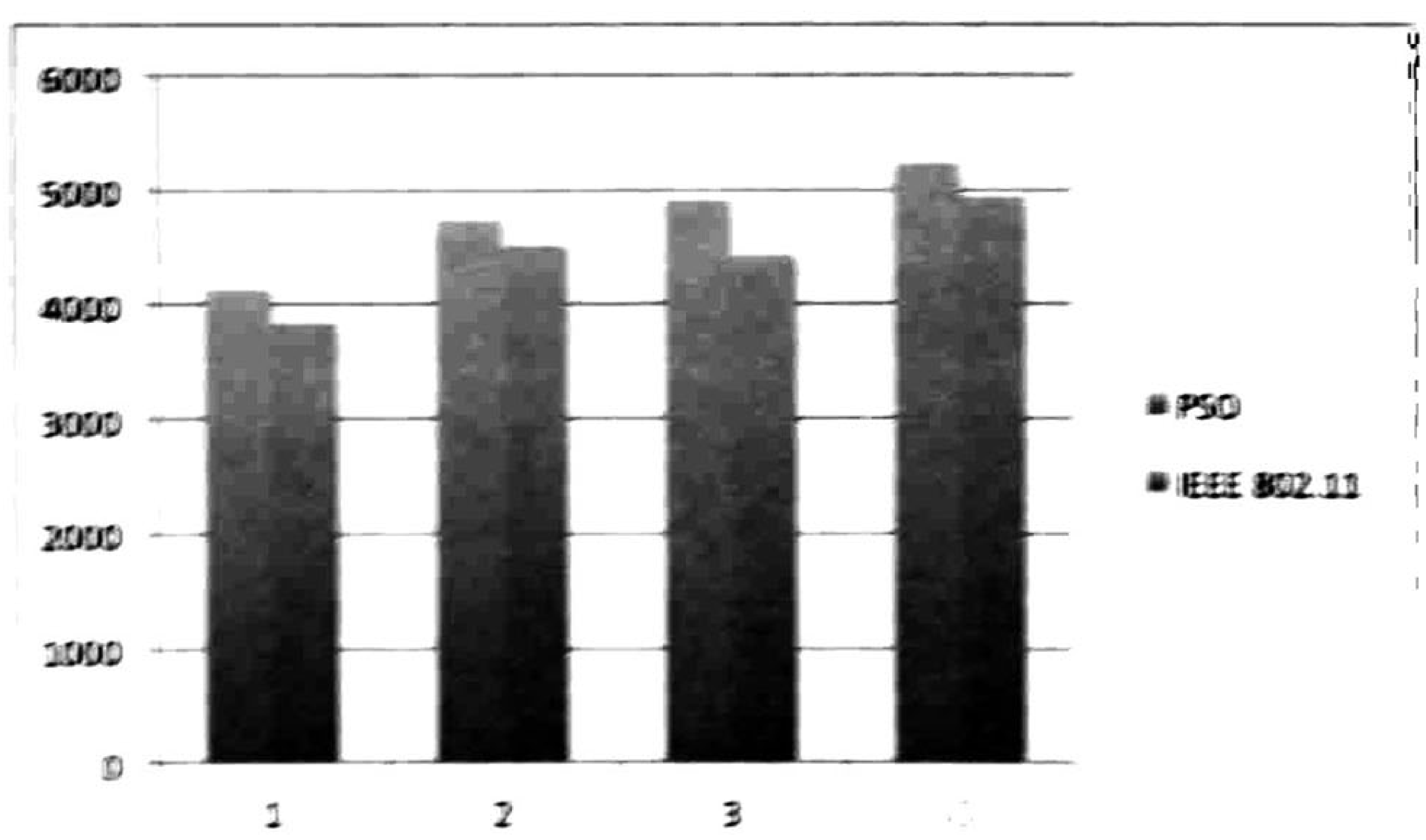


Figure 5.11: Results with 100 nodes.

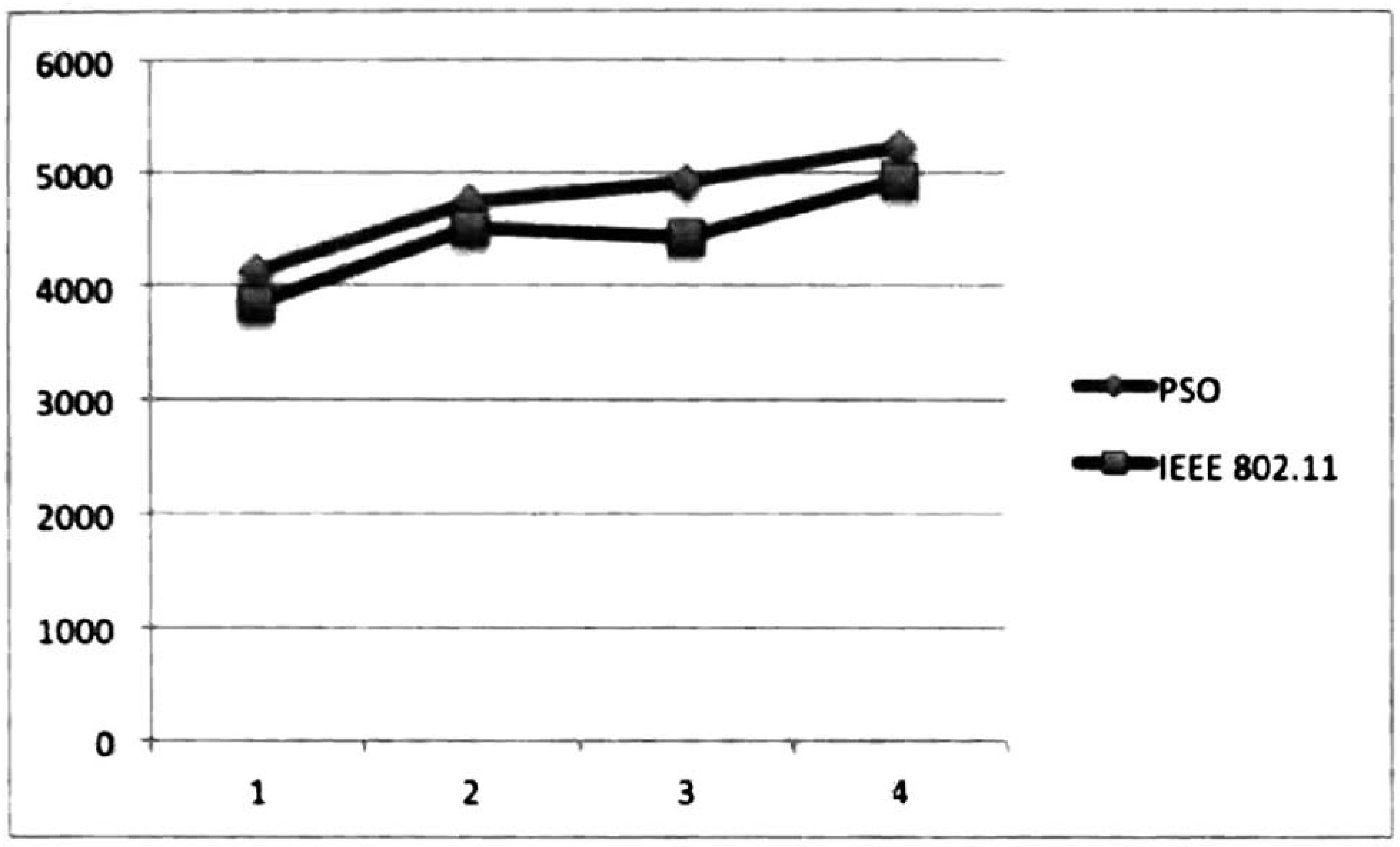


Figure 5.12: Results with 100 nodes.

In the (Table 5.4) can be observed results of the implementation against the 802.11 standard simulations.

| Test Cases | PSO | IEEE 802.11 |
|---|---|---|
| 1 | 4128 | 3840 |
| 2 | 4734 | 4510 |
| 3 | 4912 | 4431 |
| 4 | 5226 | 4937 |

Table 5.4: Comparative between Standard and PSO with 100 nodes

# Chapter 6

# Conclusions and Future Work

In this thesis, a novel Cross Layer Design based on a Multi-Objective function model was presented. This Multi-Objective model allows the representation of the different layers in the 802.11 protocol under specific constraints. This Multi-Objective function model naturally maps the ideas of cross layer design through the concept of crossing variables, which were necessary to make the model ready for the optimization algorithm. Once the model is ready, the experiments show how the Cross Layer model was able to obtain a good solution for the network model.

## 6.1 Contributions

The main contribution of this thesis are three, the fist one is the optimization model of the 802.11 protocol. There exits differents models of 802.11, but the importance of this model is that it had to be carried out by the standards of the optimization theory.

The second contribution made in this work was the implementation of the proposed algorithm for optimization in the network simulator model, this include changes to the source code. this allow to validate the optimization model.

# Appendix A

# Particle Swarm Optimization matlab Code

```
% Arturo Raymundo Aviles
% Algoritmo de Optimizacion
% Particle swarm Optimization
% Multi Objective
% In this work we present a framework that optimices
% the model of the network 802.11
% 2:37 34%
% 3:18 32%
%      P(1) - Epochs between updating display, default = 100. if 0,
%             no display
%      P(2) - Maximum number of iterations (epochs) to train, default = 2000
%      P(3) - population size, default = 24
%
%      P(4) - acceleration const 1 (local best influence), default = 2
%      P(5) - acceleration const 2 (global best influence), default = 2
%      P(6) - Initial inertia weight, default = 0.9
%      P(7) - Final inertia weight, default = 0.4
%      P(8) - Epoch when inertial weight at final value, default = 1500
%      P(9)- minimum global error gradient,
%                 if abs(Gbest(i+1)-Gbest(i)) < gradient over
%                 certain length of epochs, terminate run, default = 1e-25
%      P(10)- epochs before error gradient criterion terminates run,
%                 default = 150, if the SSE does not change over 250 epochs
```

```
%                                       then exit
%       P(11)- error goal, if NaN then unconstrained min or max, default=NaN
%       P(12)- type flag (which kind of PSO to use)
%                   0 = Common PSO w/intertia (default)
%                   1,2 = Trelea types 1,2
%                   3   = Clerc's Constricted PSO, Type 1"
%       P(13)- PSOseed, default=0
%                 = 0 for initial positions all random
%                 = 1 for initial particles as user input

function [OUT,varargout]=pso3b(functname,D,varargin)

x =[ 9 1 ;23 34; 16 28 ;12 33 ;5 7; 9 4 ;12 34 ;5 14 ;43 6 ;3 6 ;12 9; 2 30
plot (x)

t = linspace(0,2*pi);
y = sin(2*pi*t);
x = cos(2*pi*t);
plotini=0
plot(x,y)
rand('state',sum(100*clock))
disp(sprintf('the numbre of %d\n'  nargin));

if nargin < 2
   error('Not arguments enough.');
end


if nargin == 2    %]
   VRmin=ones(D,1)*-100
   VRmax=ones(D,1)*100
   VR=[VRmin,VRmax]
   minmax = 0;
   P = []
   mv = 4
   plotfcn='goplotpso'
else
   error('Wrong Number of input arguments!!!');
```

```
end


Pdef = [100 2000 1050 2 2 0.9 0.4 1500 1e-25 250 NaN 0 0];
Plen = length(P)
P    = [P,Pdef(Plen+1:end)]

df      = P(1)
me      = P(2);
ps      = P(3) %number of particles
ac1     = P(4);
ac2     = P(5)
iw1     = P(6);
iw2     = P(7);
iwe     = P(8)
ergrd   = P(9)
ergrdep = P(10)
errgoal = P(11);
trelea  = P(12)
PSOseed = P(13)




% checking errorr
 if ((minmax==2) & isnan(errgoal))
     error('minmax= 2, errgoal= NaN: choose an error goal or set minmax to 0 or
 end

 if ( (PSOseed==1) & ~exist('PSOseedValue') )
     error('PSOseed flag set but no PSOseedValue was input');
 end

 if exist('PSOseedValue')
     tmpsz=size(PSOseedValue);
     if D < tmpsz(2)
         error('PSOseedValue column size must be D or less');
     end
```

```
     if ps < tmpsz(1)
         error('PSOseedValue row length must be # of particles or less');
     end
 end

% set plotting flag  poner a 1
if (P(1))~=0
  plotflg=1;
else
  plotflg=0;
end

% preallocate variables for speed up
% variable
 tr = ones(1,me)*NaN;

% Cuidado of setting max velocity and position params
if length(mv)==1
 velmaskmin = -mv*ones(ps,D)      % min vel, psXD matrix
 velmaskmax = mv*ones(ps,D)       % max vel
elseif length(mv)==D
 velmaskmin = repmat(forcerow(-mv),ps,1) % min vel
 velmaskmax = repmat(forcerow( mv),ps,1) % max vel
else
 error('Max vel must be either a scalar or same length as prob dimension D')
end
posmaskmin  = repmat(VR(1:D,1)',ps,1)  % min pos, psXD matrix
posmaskmax  = repmat(VR(1:D,2) ,ps,1)  % max pos
posmaskmeth = 3; % 3=bounce method (see comments below inside epoch loop)

% PLOTTIN
 message = sprintf('PSO: %%g/%g iterations, GBest = %%20.20g.\n',me);

% INITIALIZE
% INITIALIZ!!!!!!!

  pos(1:ps,1:D) = normmat(rand([ps,D]),VR',1);
```

```
  if PSOseed == 1
    tmpsz                        = size(PSOseedValue);
    pos(1:tmpsz(1),1:tmpsz(2)) = PSOseedValue;
  end


  vel(1:ps,1:D) = normmat(rand([ps,D]),[forcecol(-mv),forcecol(mv)]',1);

% pbest is initial to positions vals
 pbest = pos;

% VECTORIZE THIS, or at least vectorize cost funct call
 out = feval(functname,pos);  % returns column of cost values (1 for each particle)
%---------------------------

 pbestval=out;   % initially, pbest is same as pos

% assign initial gbest here also (gbest and gbestval)
 if minmax==1
   % this picks gbestval when we want to maximize the function
    [gbestval,idx1] = max(pbestval);
 elseif minmax==0
   % this works for straight minimization
    [gbestval,idx1] = min(pbestval);
 elseif minmax==2
   % this works when you know target but not direction you need to go
   % good for a cost function that returns distance to target that can be either
   % negative or positive (direction info)
    [temp,idx1] = min((pbestval-ones(size(pbestval))*errgoal) ^2);
    gbestval     = pbestval(idx1);
 end

 % preallocate a variable to keep track of gbest for all iters
 bestpos         = zeros(me,D+1)*NaN;
 gbest           = pbest(idx1,:);  % this is gbest position
   % used with trainpso, for neural net training
   % assign gbest to net at each iteration, these interim assignments
   % are for plotting mostly
```

```
    if strcmp(functname,'pso_neteval')
        net=setx(net,gbest);
    end
 %tr(1)          = gbestval;       % save for output
 bestpos(1,1:D) = gbest;

% this part used for implementing Carlisle and Dozier's APSO idea
% slightly modified, this tracks the global best as the sentry whereas
% their's chooses a different point to act as sentry
% see "Tracking Changing Extremea with Adaptive Particle Swarm Optimizer"
% part of the WAC 2002 Proceedings, June 9-13, http://wacong.com
 sentryval = gbestval;
 sentry    = gbest;

if (trelea == 3)
% calculate Clerc's constriction coefficient chi to use in his form
 kappa   = 1; % standard val = 1, change for more or less constriction
 if ( (ac1+ac2) <=4 )
     chi = kappa;
 else
     psi     = ac1 + ac2;
     chi_den = abs(2-psi-sqrt(psi^2 - 4*psi));
     chi_num = 2*kappa;
     chi     = chi_num/chi_den;
 end
end

% INITIALIZE END INITIALIZE END INITIALIZE END INITIALIZE END
rstflg = 0; % for dynamic environment checking
% start PSO iterative procedures
 cnt    = 0; % counter used for updating display according to df in the opti
 cnt2   = 0; % counter used for the stopping subroutine based on error conve
 iwt(1) = iw1;
 % start epoch loop (iterations)
 %
for i=1:me
     pantalla= get(0,'Screensize');
     out         = feval(functname,[pos;gbest])
```

```
    y_pareto=pos(:,1); %obtengo  column 1
    x_pareto=pos(:,2); %obtengo  column 2
    %graficamos la frontera de pareto
  %  if plotini==0
  %     hplot = figure()
  %     plot(x_pareto,y_pareto,'*');
  %     plotini=plotini+1;

  %  else
  %      figure(hplot)
  %      plot(x_pareto,y_pareto,'*'); %Markersize
  %      figure(hplot+1)
  %  end
%    figure(1)
   % hold on;
    y_paret=gbest(:,1); %obtengo la column 1
    x_paret=gbest(:,2); %obtengo la column 2
   % plot3(pbest(:,1),pbest(:,D),pbestval,'g.  'Markersize',7);
   % plot(x_pareto,y_pareto,'+');
   % hold off;
   % plot(pos,feval(functname,[pos;gbest]))
    outbestval = out(end,:)
    out        = out(1:end-1,:)

    tr(i+1)           = gbestval % keep track of global best val
    te                = i % returns epoch number to calling program when done
    bestpos(i,1:D+1) = [gbest,gbestval]

    %assignin('base' 'bestpos',bestpos(i,1:D+1));
  %-----------------------------------------------------------------------
  % this section does the plots during iterations
   if plotflg==1
     if (rem(i,df) == 0 ) | (i==me) | (i==1)
        fprintf(message,i,gbestval);
        cnt = cnt+1; % count how many times we display

        eval(plotfcn); % we use that for plotting
```

```
     end  % end update display every df if statement
   end % end plotflg if statement

   % check for an error space that changes wrt time/iter
   % threshold value that determines dynamic environment
   % sees if the value of gbest changes more than some threshold value
   % for the same location
   chkdyn = 1;
   rstflg = 0; % for dynamic environment checking

   if chkdyn==1
    threshld = 0.05;  % percent current best is allowed to change, .05 = 5%
    letiter  = 5;
    % # of iterations before checking environment, leave at least 3 so PSO
    outorng  = abs( 1- (outbestval/gbestval) ) >= threshld;
    samepos  = (max( sentry == gbest ));

    if (outorng & samepos) & rem(i,letiter)==0
        rstflg=1;
      % disp('New Environment: reset pbest, gbest, and vel');
      %% reset pbest and pbestval if warranted
%       outpbestval = feval( functname,[pbest] );
%       Poutorng    = abs( 1-(outpbestval./pbestval) ) > threshld;
%       pbestval    = pbestval.*~Poutorng + outpbestval.*Poutorng;
%       pbest       = pbest.*repmat(~Poutorng,1,D) + pos.*repmat(Poutorng,1

       pbest     = pos; % reset personal bests to current positions
       %%
       %%
       pbestval  = out;
       vel       = vel*10; % agitate particles a little (or a lot)

      % recalculate best vals
       if minmax == 1
          [gbestval,idx1] = max(pbestval);
       elseif minmax==0
          [gbestval,idx1] = min(pbestval);
       elseif minmax==2 % this section needs work
```

```
        [temp,idx1] = min((pbestval-ones(size(pbestval))*errgoal).^2);
        gbestval    = pbestval(idx1);
     end

     gbest  = pbest(idx1,:);

     % used with trainpso, for neural net training
     % assign gbest to net at each iteration, these interim assignments
     % are for plotting mostly
     if strcmp(functname,'pso_neteval')
        net=setx(net,gbest);
     end
  end  % end if outorng

  sentryval = gbestval;
  sentry    = gbest;

end % end if chkdyn

% find particles where we have new pbest, depending on minmax choice
% then find gbest and gbestval
 %[size(out),size(pbestval)]
if rstflg == 0
 if minmax == 0
    [tempi]             = find(pbestval>=out); % new min pbestvals
    pbestval(tempi,1)   = out(tempi);    % update pbestvals
    pbest(tempi,:)      = pos(tempi,:); % update pbest positions

    [iterbestval,idx1] = min(pbestval);

    if gbestval >= iterbestval
        gbestval = iterbestval;
        gbest    = pbest(idx1,:);
        % used with trainpso, for neural net training
        % assign gbest to net at each iteration, these interim assignments
        % are for plotting mostly
         if strcmp(functname,'pso_neteval')
            net=setx(net,gbest);
```

```
        end
    end
  elseif minmax == 1
    [tempi,dum]         = find(pbestval<=out); % new max pbestvals
    pbestval(tempi,1)   = out(tempi,1); % update pbestvals
    pbest(tempi,:)      = pos(tempi,:); % update pbest positions

    [iterbestval,idx1] = max(pbestval);
    if gbestval <= iterbestval
        gbestval = iterbestval;
        gbest    = pbest(idx1,:);
        % used with trainpso, for neural net training
        % assign gbest to net at each iteration, these interim assignmen
        % are for plotting mostly
         if strcmp(functname,'pso_neteval')
            net=setx(net,gbest);
         end
    end
  elseif minmax == 2  % this won't work as it is, fix it later
    egones             = errgoal*ones(ps,1); % vector of errgoals
    sqrerr2            = ((pbestval-egones) ^2);
    sqrerr1            = ((out-egones) ^2);
    [tempi,dum]        = find(sqerr1 <= sqrerr2); % find particles closes
    pbestval(tempi,1) = out(tempi,1); % update pbestvals
    pbest(tempi,:)     = pos(tempi,:); % update pbest positions

    sqrerr             = ((pbestval-egones) ^2); % need to do this to ref
    [temp,idx1]        = min(sqrerr);
    iterbestval        = pbestval(idx1);

    if (iterbestval-errgoal)^2 <= (gbestval-errgoal)^2
       gbestval = iterbestval;
       gbest    = pbest(idx1,:);
       % used with trainpso, for neural net training
        % assign gbest to net at each iteration, these interim assignmen
        % are for plotting mostly
         if strcmp(functname,'pso_neteval')
            net=setx(net,gbest);
```

```
        end
    end
  end
end
%Begin of the setings

  % get new velocities, positions (this is the heart of the PSO algorithm)
  % each epoch get new set of random numbers
   rannum1 = rand([ps,D]); % for Trelea and Clerc types
   rannum2 = rand([ps,D]);
   if     trelea == 2
    % from Trelea's paper, parameter set 2
     vel = 0.729.*vel.                                % prev vel
           +1.494.*rannum1.*(pbest-pos).              % independent
           +1.494.*rannum2.*(repmat(gbest,ps,1)-pos); % social
   elseif trelea == 1
    % from Trelea's paper, parameter set 1
     vel = 0.600.*vel.                                % prev vel
           +1.700.*rannum1.*(pbest-pos).              % independent
           +1.700.*rannum2.*(repmat(gbest,ps,1)-pos); % social
   elseif trelea ==3
    % Clerc's Type 1" PSO
     vel = chi*(vel...                                % prev vel
           +ac1.*rannum1.*(pbest-pos).                % independent
           +ac2.*rannum2.*(repmat(gbest,ps,1)-pos)) ; % social
   else
    % common PSO algo with inertia wt
    % get inertia weight, just a linear funct w.r.t. epoch parameter iwe
     if i<=iwe
        iwt(i) = ((iw2-iw1)/(iwe-1))*(i-1)+iw1;
     else
        iwt(i) = iw2;
     end
    % random number including acceleration constants
     ac11 = rannum1.*ac1;    % for commonn PSO w/inertia
     ac22 = rannum2.*ac2;

     vel = iwt(i) *vel.                               % prev vel
```

```
        +ac11.*(pbest-pos)..                        % independent
        +ac22.*(repmat(gbest,ps,1)-pos);            % social
end

% limit velocities here using masking
 vel = ( (vel <= velmaskmin).*velmaskmin ) + ( (vel > velmaskmin) *ve
 vel = ( (vel >= velmaskmax).*velmaskmax ) + ( (vel < velmaskmax) *ve

% update new position (PSO algo)
 pos = pos + vel;

% position masking, limits positions to desired search space
% method: 0) no position limiting, 1) saturation at limit,
%         2) wraparound at limit   3) bounce off limit
 minposmask_throwaway = pos <= posmaskmin;  % these are psXD matrices
 minposmask_keep      = pos >  posmaskmin;
 maxposmask_throwaway = pos >= posmaskmax;
 maxposmask_keep      = pos <  posmaskmax;

 if     posmaskmeth == 1
  % this is the saturation method
   pos = ( minposmask_throwaway *posmaskmin ) + ( minposmask_keep.*po
   pos = ( maxposmask_throwaway *posmaskmax ) + ( maxposmask_keep.*po
 elseif posmaskmeth == 2
  % this is the wraparound method
   pos = ( minposmask_throwaway.*posmaskmax ) + ( minposmask_keep.*po
   pos = ( maxposmask_throwaway.*posmaskmin ) + ( maxposmask_keep.*po
 elseif posmaskmeth == 3
  % this is the bounce method, particles bounce off the boundaries wi
   pos = ( minposmask_throwaway *posmaskmin ) + ( minposmask_keep.*po
   pos = ( maxposmask_throwaway *posmaskmax ) + ( maxposmask_keep.*po

   vel = (vel.*minposmask_keep) + (-vel.*minposmask_throwaway);
   vel = (vel.*maxposmask_keep) + (-vel.*maxposmask_throwaway);
 else

 end
```

```
    %END OF
% check for stopping criterion based on speed of convergence to desired
   % error
    tmp1 = abs(tr(i) - gbestval);
    if tmp1 > ergrd
       cnt2 = 0;
    elseif tmp1 <= ergrd
       cnt2 = cnt2+1;
       if cnt2 >= ergrdep
         if plotflg == 1
          fprintf(message,i,gbestval);
          disp(' ');
          disp(['--> Probably Solution   GBest hasn''t changed by at least    ..
              num2str(ergrd),' for '  num2str(cnt2),  epochs.']);
          eval(plotfcn);
         end
         break
       end
    end

   % this stops if using constrained optimization and goal is reached
    if ~isnan(errgoal)
     if ((gbestval<=errgoal) & (minmax==0)) | ((gbestval>=errgoal) & (minmax==1))

         if plotflg == 1
             fprintf(message,i,gbestval);
             disp(' ');
             disp(['--> Error Goal reached, successful termination!']);

             eval(plotfcn);
         end
         break
     end

    % this is stopping criterion for constrained from both sides
     if minmax == 2
       if ((tr(i)<errgoal) & (gbestval>=errgoal)) | ((tr(i)>errgoal) ..
               & (gbestval <= errgoal))
```

```
        if plotflg == 1
            fprintf(message,i,gbestval);
            disp(' ');
            disp(['--> Error Goal reached, successful termination!'])

            eval(plotfcn);
        end
        break
      end
    end % end if minmax==2
   end  % end ~isnan if

%    % convert back to inertial frame
%     pos = pos - repmat(gbestoffset,ps,1);
%     pbest = pbest - repmat(gbestoffset,ps,1);
%     gbest = gbest + gbestoffset;


end  % end epoch loop

%% clear temp outputs
% evalin('base','clear temp_pso_out temp_te temp_tr;');

% output & return
 OUT=[gbest';gbestval]

 varargout{1}=[1:te];
 varargout{2}=[tr(find(~isnan(tr)))];

 return
```

# Bibliography

[1] Jaehyuk Choi, Kihong Park and Chong-kwon Kim Analysis of Cross-Layer Interaction in Multirate 802.11 WLANs" IEEE TRANSACTIONS ON MOBILE COMPUTING, P. 682-693 VOL. 8, NO. 5, MAY 2009

[2] Xiaojun Lin, Member, IEEE, Ness B. Shroff, Senior Member, IEEE, and R. Srikant, Fellow, IEEE "A Tutorial on Cross-Layer Optimization in Wireless Networks" IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 24, NO. 8, AUGUST 2006

[3] Vineet Srivastava, Institute for Infocomm Research and National University of Singapore Mehul Motani, National University of Singapore Cross-Layer Design: A Survey and the Road Ahead IEEE Communications Magazine December 2005.

[4] Xin Liu, Edwin K. P. Chong, and Ness B. Shroff Opportunistic transmission scheduling with resource-sharing constraints in wireless networks IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 19, NO. 10, OCTOBER 2001

[5] Kiran M and G. Ram Mohana Reddy Cross Layer Service Driven Adaptive Retry Limit for IEEE 802.11 Mobile Ad-Hoc Networks. IEEE 2011

[6] Yang Song, Xiaoyan Zhu, Yuguang Fang, and Hailin Zhang "Threshold Optimization for Rate Adaptation Algorithms in IEEE 802.11 WLANs" IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 9, NO. 1, JANUARY 2010.

[7] A. Goldsmith and S. B. Wicker, Design challenges for energy-constrained ad hoc wireless networks, IEEE Wireless Communications Magazine, Vol. 9, No. 4, pp. 8-27, Aug. 2002.

[8] R. Madan, S. Cui, S. Lall, and A. Goldsmith, Cross-Layer Design for Lifetime Maximization in Interference-limited Wireless Sensor Networks, IEEE Transactions on Wireless Communications, Vol. 5, No. 11, pp. 3142-3152, November 2006

[9] Book "Wireless Communications" Andrea Goldsmith, Stanford University

[10] Leticia Cagnina, Susana Esquivel and Carlos A. Coello "A Particle Swarm Optimizer for Multi-Objective Optimization" JCS T Vol. 5 No. 4 2005.

[11] J. Knowles and D. Corne. Approximating the non-dominated front using the pareto archived evolution strategy. Evolutionary Computation, 8(2):149-172, 2000.

[12] C. Fonseca and P. Fleming. Genetic algorithms for multiobjective optimization Formulation, discussion and generalization. In Fifth International Conference on Genetic Algorithms, pages 416-423, San Mateo, California, 1993. Morgan Kaufmann Publishers.

[13] author=Yayu Gao and Xinghua Sun and Lin Dai, journal=Wireless Communications, IEEE Transactions on, title=Throughput Optimization of Heterogeneous IEEE 802.11 DCF Networks,

[14] Osyczka, "Multicriteria optimization for engineering design," inDesign Optimization, J. S. Gero, ed. Academic Press: New York, 1985, pp. 193-227.

[15] James Kennedy and Russell Eberhart. "Particle swarm optimization." In Proceedings of IEEE International Conference on Neural Networks, pages 1942-1948, Piscataway, NJ, USA, 1995. IEEE Press.

[16] Z. Lu, Z. Hou and J. Du, Particle swarm optimization with adaptive mutation, Acta Electronica Sinica, vol. 32, no. 3, (2004), pp. 416-420.

[17] Haupt, R. L., and Haupt, S. E. 2004. Practical Genetic Algorithms. Second Edition, Wiley Interscience.

[18] S. Cui, R. Madan, A. Goldsmith, S. Lall, Joint routing, MAC, and link layer optimization in sensor networks with energy constraints, in Proc. IEEE ICC 05, vol. 2, pp. 725 729, May 2005.

[19] Poli, R. 2007. Analysis of the publications on the applications of particle swarm optimization applications. Tech. Rep. CSM-469, Department of Computing and Electronic Systems, University of Essex, Colchester, Essex, UK.

[20] Steuer, R.E. (1986). Multiple Criteria Optimization: Theory, Computations, and Application. New York: John Wiley and Sons, Inc

[21] Massimiliano Laddomada and Fabio Mesiti. On the Optimization of the IEEE 802.11 DCF: A Cross-Layer Perspective Hindawi Publishing Corporation International Journal of Digital Multimedia Broadcasting Volume 2010, Article ID 397824, 13 pages doi:10.1155/2010/397824

[22] Yi-Hung Huang1 and Chao-Yu Kuo Dynamic tuning of the IEEE 802.11 distributed coordination function to derive a theoretical throughput limit Huang and Kuo EURASIP Journal on Wireless Communications and Networking 2011 http://jwcn.eurasipjournals.com/content/2011/1/105

[23] QoS Support in Wireless Sensor Networks: A survey, Dazhi Chen and Pramod K. Varshney

[24] Terms and Definitions Related to Quality of Service and Network Performance Including Dependability, ITU-T Rec. E.800 (0894), 1994 (ITU).

[25] Mangold, S.; Sunghyun Choi; Hiertz, G.R.; Klein, O.; Walke, B., "Analysis of IEEE 802.11e for QoS support in wireless LANs," Wireless Communications, IEEE , vol.10, no.6, pp.40,50, Dec. 2003

[26] An Improved Ant Colony Optimization Algorithm for Multiple QoS Routing to find multiple feasible paths for packet switched Network Meenakshi R Patel, Babita Kubde Rungta College of Engineering and Technology, Bhilai,C.G., India patelmeenu@gmail.com Meenakshi R Patel et al ,Int.J.Computer Technology and Applications,Vol 3 (3), 1150-1152 ISSN:2229-6093 International Journal of Computer Technology and Applications.

[27] Hongqiang Zhai; Chen, X.; Yuguang Fang, "How well can the IEEE 802.11 wireless LAN support quality of service?," Wireless Communications, IEEE Transactions on , vol.4, no.6, pp.3084,3094, Nov. 2005.

[28] Felemban, E.; Ekici, E., "Single Hop IEEE 802.11 DCF Analysis Revisited: Accurate Modeling of Channel Access Delay and Throughput for Saturated and Unsaturated Traffic Cases," Wireless Communications, IEEE Transactions on , vol.10, no.10, pp.3256,3266, October 2011

[29] Pourmohammad, S.; Soosahabi, R.; Perkins, D.; Fekih, A, "An analytical QoS model for IEEE 802.11-based single and multihop wireless networks," Computing, Networking and Communications (ICNC), 2014 International Conference on , vol., no., pp.914,920, 3-6 Feb. 2014

[30] Giuseppe Bianchi, Luigi Fsatta, Matteo Oliveri "Performance Evaluation and Enhancement of the CSMA/CA protocol for 802.11 wireless LANs"

[31] Giuseppe Bianchi Performance Analysis of the 802.11 Distributed Coordination Function, : Journal on Selected Areas in Communications, March 2000 Volume: 18,Issue: 3

[32] Duffy, K.. Malone, D. Leith D. J. Modeling the 802.11 distributed coordination function in non-saturated conditions, IEEE Communications Letters, 9 (8). pp. 715-717. ISSN 1089-7798

[33] Malone, D. Duffy, K., Leith, D. Modeling the 802.11 Distributed Coordination Function in Nonsaturated Heterogeneous Conditions presented in ACM Transactions on Networking. Vol 15, No. 1 February 2007

[34] IEEE 802.11 WG, Reference number ISO/IEC 8802-11:1999(E) IEEE Std 802.11, 1999 edition. International Standard Information Technology - Telecommunications and information exchange between systems-Local and metropolitan area networks Specific Requirements - Part 11:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999. IEEE 802.11 WG, Draft Supplement to STANDARD FOR Telecommunications and Information.

[35] 802.11n-2009 "IEEE Standard for Information technology Local and metropolitan area networks" Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC)and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput.

[36] Wireless LAN at 60 GHz IEEE802.11ad at http:www.ieee802.11ad/standar

[37] Sebastian Zander, Thuy Nguyen, Grenville Armitage *Automated Traffic Classification and Application Identification using Machine Learning.* The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05).

[38] Isara Anantavrasilp, Thorsten Scholer "Automatic flow classification using machine learning" 15th International Conference on In Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007. 15th International Conference on (2007), pp. 1-6.

[39] Rafael del-Hoyo et al Neural Networks for QoS Network Management IWANN 2007, LNCS 4507, pp. 887-894, 2007.

[40] D. Qiao, S. Choi and K. G. Shin. Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs. IEEE Transactions on Mobile Computing. Vol. 1. No. 4, pp. 278-292.

[41] Moses Ekpenyong and Joseph Isabona "Modeling Throughput Performance in 802.11 WLAN" IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 11, May 2010

[42] P. Cheeseman, J. Stutz, Bayesian Classification (Autoclass): Theory and Results, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, USA, 1996.

[43] Dr. Fayez Gebali "Analysis of Computer and Communication Networks" ISBN: 978-0-387-74436-0 2008 Springer

[44] T. S. Kim, H. Lim, and J. C. Hou. Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. In In Proc of the ACM Mobicom, 2006.

# 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

Optimización Multi-Objetivo para calidad de servicio en redes inalámbricas IEEE 802.11 / Multi-Objective Optimization for Quality of Service in IEEE 802.11 wireless networks

del (la) C.

Arturo RAYMUNDO AVILÉS

el día 05 de Diciembre de 2014.

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

Dra. Ofelia Begovich Mendoza
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González Pico
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dra. Susana Ortega Cisneros
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

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

Dr. Luis Rizo Domínguez
Profesor
Instituto Tecnológico y de Estudios Superiores de Occidente