الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي

Ministry of Higher Education and Scientific Research

Nº Ref: ……..

Abdelhafid Boussouf
University Center of Mila

Institute of Science and Technology

Sector: Computer science

Thesis Presented In view of obtaining the Master's degree

Field: Mathematics and Computer Science

Sector: Computer science

Specialty: Information and Communication Sciences and Technologies (STIC)

A security assistant for Smart Cities IoT Infrastructure

By :

GHODBANE Intissar

FEDALA Naoual

Before the Jury :

President	ZEKIOUK Mounira	U.C. Mila
Supervisor	LALOUCI Ali	U.C. Mila
Examiner	BOUSSOF Hakim	U.C. Mila

Years: 2022/2023

شكر و تقدير

أول من يشكر ويحمد اناء الليل واطراف النهار هو العلي القهار ،الذي اغرقنا بنعمه التي لاتعد ولا تحصى
فالحمد لله الذي أتم علينا نعمه و والى علينا مننه وأعاننا لإنجاز هذه المذكرة
وتقديرا منا وإعترافاً بالجميل نتقدم بجزيل الشكر لأولئك الذين لم يألوا جهداً في مساعدتنا في مجال البحث العلمي، وأخص بالذكر الأستاذ الفاضل: **علي لالوسي** على هذه الدراسة وصاحب الفضل في توجيهنا ومساعدتنا في انجاز هذا العمل, فجزاه الله كل خير.
كما نشكر كل اساتذة قسم الرياضيات والاعلام الآلي ونخص بالذكر الاستاذ "بن الشيخ لحسين ماجد"

والشكر موصول الى كل معلم افادنا بعلمه من أولى المراحل الدراسية حتى هذه اللحظة.

و في الأخير ندعو الله أن يرزقنا السداد والرشاد وأن نواصل السير في درب العلم ونكون رمزا للهمة فخرا للأمة.
و الحمد لله أولا وآخرا

الإهداء

الحمد لله وما توفيقي الا به، وبعد فاني أهدي عملي هذا:

الى من كلله الله بالهيبة والوقار .. الى من أحمل اسمه بكل افتخار، أبي العزيز أدامك الله لي

الى من حملتني وهنا على وهن....الى معنى الحب والحنان ، أمي الغالية حفظك الله ورعاك

الى أجمل ما أهداني القدر.. أختي مرام وأختي التي لم تلدها أمي مريم

الى شقيقي الغاليين.. أيمن وتقي الدين

الى صديقاتي ورفيقات دربي، صفاء، لمياء، سارة، نوال

الى أسرتي الكبيرة، غضبان، ناجي

الى كل من ساعدني من قريب او بعيد لإنجاز هذا العمل.

غضبان انتصار

الاهداء

الحمد لله وكفى والصلاة والسلام على المصطفى مُحَمَّد وآله وصحبه ومن اقتفى وبعد فاني أهدي ثمرة مجهودي هذا:

الى كل طالب يقول يوم تخرجه ليتك معي أمي ، اكتب اهدائي اليك وليتك تقرئين ،نور الله مرقدك وعطر مشهدك وطيب مضجعك وجعل الجنة مثواك.

الى من علمني الكفاح كي أذوق طعم النجاح الى أبتي رزقني الله مرضاته وبره وادامه نورا لدربي.

الى بؤرة النور التي روضت سنوات اليتم واستمرت نبعا لا يعرف النضوب ، أختي الكبرى.

الى ملاكي في الحياة ورفيق الدرب وسندي في الايام كلها حلوها ومرها ،زوجي.

الى من بهم يشد ساعدي وتعلى هامتي هم سندي وركائز نجاحي ، اخواتي واخوتي وأطفالهم وكل عائلتي.

الى أسرتي الثانية، من كانت أجمل ذكرياتي معهم رفيقاتي، وأخص بالذكر انتصار رفيقة المشوار.

الى كل من ذكرهم قلبي ونسيهم قلمي.

فضالة نوال

Abstract

As the world population is moving towards big cities, and these cities need to find new ways to endorse high service levels for citizens which includes smart metering, smart grids, smart parking, optimized driving and walking routes, etc. This smart cities use IoT devices (a.k.a. IoT nodes) such as connected sensors, lights, and meters to collect and analyze data. In IoT networks dedicated to smart-cities, not all nodes have the same importance, and some are more important than others. A node is important if its failure or malicious behavior significantly degrades system performance. The identification of the most important nodes in networks has long been the subject of extensive investigation. Depending on their function in the system, these nodes go by several names in the literature, including: most influential nodes, critical node, independent nodes, dominating nodes, secure dominating nodes, etc.

We have conducted a study on the problems of CNDP that need the MIS and using both sequential and distributed approaches.

The objective of this project is to propose a solution that assists administrators in the design and securing of smart cities in order to reduce the cost of installing different devices.

Key words:

IoT Security, IoT Simulation, Cupcarbon simulator, WBS concept, Smart cities, Algerian cities, Maximal Independent Set, CNDP problem, Distributed algorithm.

الملخص

مع انتقال سكان العالم إلى المدن الكبرى ، يجب أن تجد هذه المدن طرقًا جديدة للقلق بشأن مستويات الخدمة العالية للمواطنين ، بما في ذلك العدادات الذكية والشبكات الذكية ومواقف السيارات الذكية وطرق القيادة والخطوات المحسّنة ، إلخ. تستخدم هذه المدن أجهزة إنترنت الأشياء الذكية (المعروفة أيضًا باسم عقد إنترنت الأشياء) مثل أجهزة الاستشعار المتصلة والأضواء والعدادات لجمع البيانات وتحليلها. في شبكات إنترنت الأشياء المخصصة للمدن الذكية، ليست كل العقد لها نفس الأهمية، وبعضها أكثر أهمية من البعض الآخر. تعتبر العقدة مهمة إذا كان فشلها أو سلوكها يؤدي إلى تدهور أداء النظام بشكل كبير. لطالما كان تحديد أهم العقد في الشبكات موضوع تحقيق مكثف. وفقًا لوظيفتها في النظام، تحتوي هذه العقد على عدة أسماء في الأدبيات، بما في ذلك: العقد الأكثر تأثيرًا، والعقد الحرجة، والعقد المستقلة، والعقد السائدة، والعقد المهيمنة الآمنة، وما إلى ذلك

الهدف من هذا المشروع هو اقتراح حل يساعد المسؤولين في تصميم وتأمين المدن الذكية من أجل تقليل تكلفة تركيب الأجهزة المختلفة

الكلمات المفتاحية:

أمن أنترنت الأشياء، محاكي أنترنت الأشياء، محاكي كاب كربون، مفهوم الانتظار قبل البدء، المدن الذكية، المدن الجزائرية، أقصى مجموعة مستقلة، مشكلة الكشف عن العقد الحرجة، الخوارزمية الموزعة.

List of Abreviations

IoT	Internet of Things.
DAS	Data Acquisition Systems
AR	Augmented Reality
HE	Homomorphic Encryption.
SDN	Software Defined Networking.
ICT	Information Communication and Technology.
AI	Artificial Intelligence.
CNDP	Critical Node Detection Problem.
MIS	Maximal Independent Set.
3C-CNP	Component Cardinality Constrained Critical Node Problem.
WSN	Wireless Sensor Network
WBS	Wait Before Starting.
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers

List of Figures

List of Tables

List of Algorithms

Table of Content

General introduction

As the world population is moving towards big cities, and these cities need to find new ways to endorse high service levels for citizens. The number of smart city projects is increasing across the world, and it is essential to discuss what are the goals and challenges involved to make a city smarter. According to a study done by Unicef (www.unicef.org/sowc2012/urbanmap), 70% of the world's population will live in cities, by 2050 [1]. These big cities will bring some intense challenges regarding their management, but these same issues make them into engines of new research ideas and innovations. In order to convert these big cities into smart cities, we must improve its fundamental IoT infrastructure.

Internet of Things technology is emerging and emerging in recent years. It is characterized by a gigantic number of mobile and heterogeneous objects connected, which requires effective mechanisms to supervise them and ensure their security. The Internet of Things (IoT) enables global connectivity to remote intelligent devices. This technology involves the detection, communication and processing of real-time data received from billions of connected devices with minimal human intervention. Internet exposure and the constraints of IoT devices, generally limited memory limited memory, low processing capacity and mainly battery-powered operations make them vulnerable to a variety of attacks.

Traditional security models for Internet applications are not adapted to the IoT, because it's generally not real-time. Therefore, it is important to have alternative solutions that offer significant security for IoT devices/applications. Emerging technologies such as: blockchain, light cryptography, in particular authentication and confidentiality techniques (cryptographic key algorithms and others, etc.) have been proposed.

In this thesis, we propose a new solution which is based on graph parameters. Graphs have seen a resurgence of interest in modeling many problems in computer science. A graph can be used to model a program or an algorithm, but also various types of networks. The common approach for problems solving using graphs consists of: modeling the problem by a graph, searching the appropriate property (parameter) corresponding to the studied problem, and finding the parameter value using suitable algorithms.

Our thesis is organized into four chapters as follows:

The First chapter: IoT Concepts & Security

In this chapter, we will give an overlook of some basic concepts of IoT networks and we will define briefly the IoT security techniques. Moreover, we introduce the basic concepts of smart cities which are based on IoT devices.

The Second chapter: Distributed graph algorithms & CNDP

This chapter presents the concepts of graph, distributed system, distributed algorithm. And also present the problems of Maximal Independent Set and Critical Node Detection Problem with sequential and distributed approaches.

The Third chapter: Simulation and modeling

In this chapter, we will describe the simulation part which we will talk about our working environment and the modelling part where we will analyze the algorithm of 3C-CNP in different cases and with stages of implementation step by step. Finally, we will propose two WBS algorithms for MIS and 3C-CNP problems that will be used to build our simulation.

The Fourth chapter: Experimental results

This chapter is divided into two parts. The first part will present our experimentation. To do this, will choose an Algerian city as case study firstly and make the simulation. The second part will compare between our work and other work for a best clarification.

Chapter 1 IoT: Concepts & Security

Introduction

In this chapter, we will start by giving an overlook of some basic concepts of IoT networks. After that, we will define briefly the IoT security techniques. Moreover, we introduce the basic concepts of smart cities which are based on IoT devices.

1.1 IoT basic concepts

This section is devoted to introduce the basic concepts of IoT. Firstly, we give a short definition of IoT. Then we will present the architecture of IoT. After that, we will talk about application domain of IoT.

1.1.1 Definition of IoT

The Internet of Things (IoT) is a set of technologies that uses sensors and actuators to inform us about the status of everyday items such as vehicles, tools and even living beings. It allows us to interact with them, enabling connectivity with platforms in the cloud that receive and process information for posterior analysis. This analyzed data is then used to make decisions [2].

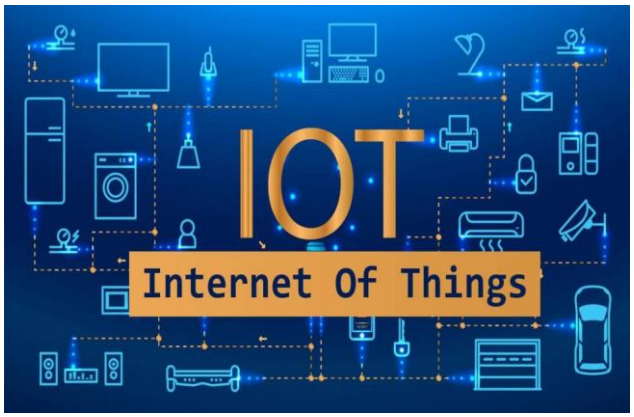

Figure1.1: IoT system

1.1.2 The architecture of the Internet of Things

There is no standardized IoT architecture, but the basic and widely accepted format is the four-layer IoT architecture [3]:

- **Perception layer**

This layer converts analog signals into digital data and vice versa. It encompasses a wide range of objects that bridge the real world and the digital world. These are classified into 3 categories:

- Sensors such as probes, gauges or counters.
- Actuators for motor control, lasers as well as robotic arms.
- Machines and devices connected to sensors and actuators.

- **Network layer**

The main function of the network layer is to connect the devices to other smart objects, servers and network devices for efficient processing of collected data. The latter also manages the transmission of all data.

We find not only in the network layer the Internet and network gateways are responsible for the connection between the sensor networks and the Internet. But also Data Acquisition Systems (DAS) which perform the data aggregation and conversion function.

- **Data processing layer**

The processing layer has 3 main missions:

- The accumulation of real-time data from the network layer to allow the administrator to define the relevance of the data and its location.
- Storing relevant and truly useful data in a wide range of storage solutions.
- Data processing to improve the interoperability of smart devices.

To perform its functions, the latter relies on IoT platforms.

- **Application Layer**

The application layer that will interact with the user through specific services. It is possible to build applications directly on IoT platforms capable of offering a software development infrastructure with turnkey tools that allow data exploration, advanced analysis and data visualization.

1.1.3 Application domain

The IoT will cover a wide range of applications and will affect almost all areas that we face on a daily basis, this will allow the emergence of intelligent spaces. Among these smart spaces include:

- **Smart Health**

In the field of health, the IoT will allow the deployment personal networks for the control and monitoring of clinical signs, in particular for elderly people, connected objects make it possible to monitor blood pressure, heart rate, quality of breathing or fat mass. This will facilitate the remote monitoring of patients at home, and provide solutions for the autonomy of disabled [3].

- **Industry**

IoT technology will allow total monitoring of products, the supply chain production, up to the logistics and distribution chain by supervising the conditions supply. This end-to-end traceability allows factories to improve efficiency of its operations, optimize production and improve the safety of employees [4].

- **Smart city**

The vision of the “connected city”, also referred to as the “digital city” or the “smart city”, foresees that the whole of the urban space will be interconnected and will interact with the digital world. This paves the way in particular for the sustainable development of our urban environments, combined with an improvement in the quality of life and security of citizens while

respecting their privacy. It is also an opportunity to review the processes of regulation and management of cities through increased citizen engagement through social networks and sensors, physical or software, embedded in smart phones [5].

- **Smart grid**

We often talk about these intelligent networks or "smart grids" about electricity networks which, thanks to computer technologies, adjust the flow of electricity between suppliers and consumers. By collecting information on the state of the network, smart grids contribute to a balance between production, distribution and consumption [6].

- **Smart agriculture**

Aims to enhance the capacity of agricultural systems, contribute to food security by integrating the need for adaptation and the potential mitigation in sustainable agriculture development strategies .This objective was finally achieved by the use of new technologies, such as satellite imagery and computing, satellite positioning systems such as GPS, also by the use of sensors which will take care of collecting useful information on the state of the soil, humidity rate, mineral salt rate, etc. And send this information to the farmer to take the necessary measures to guarantee good production [7].

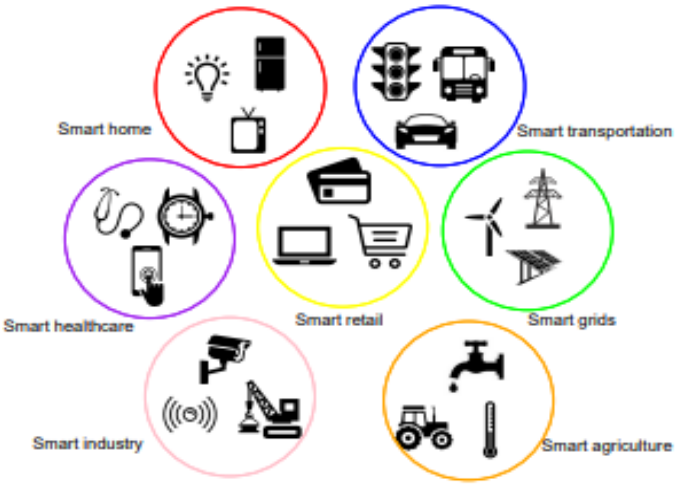

Figure1.2: Application domain of IoT

1.2 Security in the Internet of Things

This section is devoted to introduce the basic concepts of IoT Security. Firstly, we give a short definition of IoT security. Then we talk about its objectives. After that, we will end this section by the types of attaques. Finally, we will give the different security approaches used to protect these systems.

1.2.1 Definition

Computer security in general consists in ensuring that the resources an organization's hardware and software is only as intended. She lives at ensure several objectives, of which the five main ones are: Authentication, Integrity, availability and non-repudiation [8].

1.2.2 The principles of security

- **Confidentiality**

The most crucial aspect of network security is privacy. Confidentiality protects the privacy of data transmission between a sender and a recipient by guaranteeing that it is only accessible by those to whom access has been granted. The only effective method for guaranteeing data confidentiality is cryptography or data encryption.

- **Authentication**

It is the process used to confirm a node's identity before it may communicate with other nodes. The sensor node has to be able to validate the source node's identification when an attacker forges and injects faked packets into the network.

- **Integrity**

Integrity can be thought of as a set of safeguards that ensure that data is protected from unauthorized modification and alteration. Attacks against integrity aim to modify, add, or remove resources or information.

- **Availability**

Availability is a network service that provides assurance to entities authorized to access network resources. The objective is to avoid attacks of the type denied service [8].

- **Non-repudiation**

Mechanism that makes sure a message was delivered by the sender and received by the recipient and that the correspondents cannot dispute either action.

1.2.3 Classification of attacks

According to very specific criteria, such as the power of the attacker, whether or not the latter belongs to the network. Attacks against sensor networks can be classified according to the following categories [9]:

- **External attacks vs internal attacks**

An external attack occurs from outside the sensor network. They produced by nodes which are not deployed inside the network and which are not allowed to participate in the network. While insider attacks happen through nodes malicious internals.

- **Passive Attack vs Active Attack**

- Passive attacks are only interested in collecting information sensitive without any modification or influence on the communication. These informations collected as the detection of important nodes in the network (Cluster-Head) can then help the attacker to carry out malicious attacks
- Active attacks aim to disrupt the function of the network and the degradation of its performance. Attacker attempts to exploit network security vulnerabilities to launch various attacks with the aim of modifying the data.

- **Physical Attacks vs Ranged Attack**

In the physical attack, an adversary physically gains access to the sensor node which is harmed by tampering with or destroying node material. On the other hand, an attack distance is implemented from a distance, for example, by transmitting a signal at high energy to interrupt the communication.

1.2.4 Emerging solutions for security

In the literature, several solutions have been proposed to protect IoT networks. We can cite the best-known [10]:

- **Cryptography**

Encryption is the process of securing data against unauthorized access, So that only the recipients of the information can read and process it.

- **Machine Learning**

Machine learning refers to intelligent methods used to improve performance standards using sample data or experiment passed through learning.

- **Homomorphism and searchable encryption**

HE (Homomorphic Encryption) is an encryption scheme which, due to its homogeneous properties, allows computation on encrypted data and is based on a cipher asymmetric or public key encryption.

- **Fog computing**

Fog computing falls under a broader definition of Edge Computing, which consists of push applications and services, in whole or in part, to the network edge.

- **Software-Defined Networking**

The SDN separates the level network control of data transmission level. The latter is implemented by SDN switches using the "rules" defined by a central component, the SDN controller.

- **The Blockchain**

The blockchain is a distributed and decentralized ledger that allows storing and exchanging information without a trusted third party.

- **Important Nodes Detection**

Protecting applications against malicious behavior is one of the primary concerns in designing a network application. Thus, the design must take into account the weaknesses of the network that an attacker can exploit. Generally, the design of network applications is based on the selection of a set of nodes called "kernel", such as the Maximal Independent Set or the Minimal Dominating Set. Thus, their security depends on the security of this kernel. The principle that the more critical nodes there are in the kernel, the application is vulnerable, and conversely, the fewer critical nodes there is more robust the application; the CNDP is a good parameter to consider when designing secure network applications [11].

1.3 Smart cities

This section is devoted to introducing the field of our study, the smart city. Firstly, we give a short definition. Then we explain what makes a city smart and the relation between smart cities and IoT. After that, we will present how can make my city become a smart city. We will end this section by giving advantages of smart city.

1.3.1 What are smart cities?

A smart city is an economically vibrant city that provides its citizens a good quality of life using Information Communication and Technology (ICT) solutions. The Smart Cities Council 1 defines a smart city as one that uses digital technology for all city functions. Similarly, World Bank2 defines a smart city as a technology-intensive city that has sensors installed everywhere and offers highly efficient public services using information gathered in real time by thousands of interconnected devices. Further, a smart city cultivates a better relationship between citizens and governments using the available technology. It relies on feedback from citizens to help improve service delivery. It puts in place mechanisms to gather this information [12].

Figure1.3: Smart city

1.3.2 What makes a city “smart”?

A smart city uses cloud computing, Artificial Intelligence (AI), IoT, Augmented Reality (AR), edge, Blockchain, and other intelligent technologies to improve infrastructure, elevate government services, enhance quality of life, and fuel economic growth.

The world’s most successful smart cities are centered around data. These cities’ forward-thinking governments invest in and use advanced technologies to quickly collect and analyze data—whether it’s information for residents, city resources, and infrastructure, transportation, or carbon emissions. The insights that governments gain from this data enable them to make informed decisions and take steps to streamline services, engage with residents, support business growth, and drive sustainability efforts.

1.3.3 What a relation between smart cities and IoT?

Smart city governments are increasingly using IoT for two important tasks—remote monitoring and predictive maintenance. Remote monitoring allows administrators to monitor assets, such as vehicles and heavy machinery, either continuously or at regular intervals. This allows city managers to track the asset's location, performance, and condition. The insights city governments gain from this smart city technology can help them [13]:

- Decrease fuel, maintenance, and service costs.
- Refine city government processes.
- Provide a better experience for the population.
- Reduce wear and tear on vehicles and machinery.
- Increase the number of service appointments per day.
- Route vehicles more efficiently.
- Improve resident security and city employee safety.

Predictive maintenance incorporates machine learning software to analyze data, predict outcomes, and automate operations. This technology allows city managers to identify and overcome issues before they become a major problem. Predictive maintenance helps smart city administrators:

- Pinpoint mechanical or operational issues that are leading to failure or slowdowns.
- Determine which spare parts to maintain in inventory for repair issues.
- Predict and prevent equipment failures before they occur.

Governments that invest in IoT can use this smart city solution to deliver greater value to the public, connect with people in a more personalized way, reduce materials and labor waste, and boost operational efficiency [11].

1.3.4 What are the benefits of smart cities?

Smart cities use innovative smart technologies to:

- Enhance standard of living.
- Improve processes.
- Streamline the transport system.
- Fuel economic expansion.
- Provide superior services.

Conclusion

In this chapter, we have given some general information about the internet of thing such as: its architecture, its application domain. Then we have described the main concepts related to IoT security. At the end of this chapter, we have presented our application field.

In the next chapter, we will present the problem of critical node detection problem (CNDP)

Chapter 2: Distributed graph Algorithm& CNDP

Introduction

In this chapter, we will start by giving an overlook of some basic concepts of graphs. Then, we will talk about distributed system and algorithm .After that we will define briefly the maximal independent set problem. Moreover, we will present a brief survey on critical node detection problem in networking in general.

2.1 Graph Basics

This section is devoted to introduce the basic concepts of graphs theory. Firstly, we give some definition and notation used throughout this thesis. We will end this section by graph classes and graph parameters.

2.1.1 Definition and notation

Definition 2.1 (Graph)

A graph 'G' is a set of vertex, called nodes 'V' which are connected by edges, called links 'E'. Thus G= (V, E) or G (V, E) [14].

Definition 2.2 (degree of vertex)

We call the degree of the vertex v, and we noted (v), the number of edges incident to this vertex .A loop on a vertex counts double. In a simple graph, we can also define the degree of a vertex as the number of its neighbors [15].

Definition 2.3 (connected and disconnected graph)

- A graph is connected if a path connects any two vertices of the graph.
- A graph is disconnected if at least two vertices of the graph are not connected by a path [16].

Definition 2.4 (Algorithm)

An algorithm is a series of procedures used to carry out a calculation.

Definition 2.5 (Deterministic algorithm)

A deterministic algorithm is one that always yields the same output given the same input.

Definition 2.6 (Greedy algorithm)

A greedy algorithm is an algorithm that attempts to find a global optimum by making the locally optimal choice in each step according to a given rule. The general greedy algorithm for MIS is non-deterministic [17].

Definition 2.7(Distributed algorithm)

A distributed algorithm is an algorithm that will run on each node in the distributed system. A distributed algorithm is uniform if and only if all processors execute the same algorithm. An algorithm is non-uniform if at least one processor does not execute the same algorithm [18].

2.1.2 Connectivity

The connectivity is a key concept in graphs. A graph is said to be connected if it is formed from a single piece. More formally, this concept is based on the existing of paths between pair of nodes. By definition, a path is a sequence of distinct nodes such that each consecutive pair of nodes is joined by an edge. Thus, a graph is connected if there exists a path between each pair of their nodes; otherwise, the graph is disconnected. A disconnected graph consists of two or more pieces called connected components [19].

In what follows, we provide the principal concept directly related to the Connectivity

2.1.3 Subgraphs and induced Subgraphs

Let's consider a graph G = (V, E), we say that G′ = (V ′, E′) is a subgraph of G if $V' \subseteq V$ and $E' \subseteq (E \cap (V' \times V'))$, if $V' = V$ we say that G is a spanning Subgraphs of G. An induced subgraph of G denoted by G[S] or ⟨S⟩ is the maximal subgraph of G with nodes set S, i.e, $\forall(u, v) \in S \times S$, u, v are adjacent in G[S] if and only if they are adjacent in G. This, we can say that the connected components are the maximal induced connected subgraph of G [19].

Now, we provide the most well-known subgraphs.

- **Cycle:** A cycle Ci, $i \geq 3$ consists of nodes set $v_1, v_2, \ldots, v_i$ and edges set

$\{v_1v_2, v_2v_3, \ldots, v_{i-1}v_i, v_iv_1\}$.

- **Clique:** A subgraph induced by $Si = \{v_1, v_2 \ldots v_i\}$ forms the clique Qi if and only if $\{\forall vj , v_k \in Si : v_jv_k \in E\}$, i.e, the number of edges in the clique Qi is $\frac{i(i-1)}{2}$. If $Si \equiv V$ then we say that G is a complete graph.
- **Independent set:** A subgraph induced by $I = \{v_1, v_2, \ldots, v_i\}$ forms an independent set I if and only if $\{\forall v_j , v_k \in I : v_jv_k \notin E\}$, i.e, the number of edges in G[I] is 0 .

2.1.4 Graph parameters

Graph parameters are very useful in graph characterization and problems resolution. Generally, a real-life problem corresponds to a parameter in the graph modeling such one. A graph parameter is an invariant that depends only on the abstract structure of the graph (eg. number of nodes or edges) and not on the graph representation (eg. graph drawing) [19].

2.2 Distributed system

We introduce this section by a definition of distributed system, communication models and same example of distributed system .After that we talk about distributed computing platforms.

2.2.1 Definition

A distributed system is a collection of independent components located on different machines that share messages with each other in order to achieve common goals. Distributed systems meant separate machines with their own processors and memory [20].

2.2.2 Communication models

The basic models of a distributed system are the message passing and shared-memory models [20]:

- In the message passing model, nodes of the distributed system communicate by messages only. Messages are communicated in rounds in synchronous message passing, where messages sent in round *k* are delivered to all recipients before messages in round *k* + 1 can be transferred. In asynchronous message passing, however, messages are assumed to eventually reach the destinations after unknown delays. Analyzing asynchronous message passing algorithms is more difficult than synchronous ones due to the uncertainties involved.
- In shared-memory models, processes communicate by reading and writing to shared memory. Synchronization is an important issue also in shared-memory systems. *Distributed shared-memory* systems implement shared memory model over the message passing model to use the available shared memory software modules conveniently.

2.2.3 Examples of distributed system [20]

- **Telecommunication networks:** telephone and cellular networks are also examples of distributed networks. Telephone networks have been around for over a century and it started as an early example of a peer to peer network. Cellular networks are distributed networks with base stations physically distributed areas called cells
- **Distributed artificial intelligence:** distributed Artificial Intelligence is a way to use large scale computing power and parallel processing to learn and process very large data sets using multi-agents.
- **Distributed Database Systems:** a distributed database is a database that is located over multiple servers and/or physical locations. The data can either be replicated or duplicated across systems.

2.2.4 Distributed Computing Platforms

Due to the recent technological advancements, in the last few decades, we have witnessed diverse distributed system platforms such as the Grid, the Cloud, mobile ad hoc networks, and wireless sensor networks [20].

2.3 Maximal Independent Set Problem (MIS)

We talk in this section about the maximal independent set because we need it in our problem (CNDP). So, we will present a sequential algorithm and a distributed algorithm for the maximal independent set which are found in the literature.

2.3.1 Independent set

An independent set is a set of vertices such that no two vertices in the set are adjacent. An independent edge set, or matching, is a set of edges such that no two edges in the set are incident to the same vertex [17].

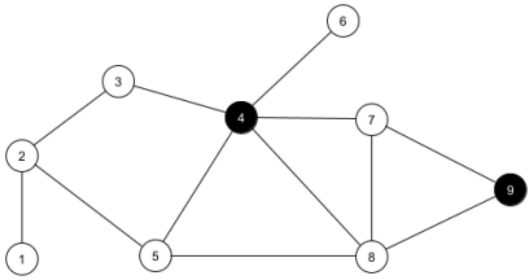

Figure 2.1: Independent Set

2.3.2 Maximal independent set

A maximum independent set (MIS) is an independent set such that no other independent sets of G are larger. The number of elements in a MIS is denoted α(G).

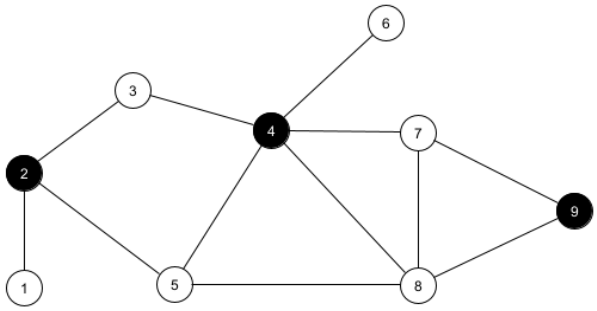

Figure2.2: Maximal Independent Set.

2.3.2 A sequential algorithm of maximal independent set

In the literature, a lot of sequential algorithms have been suggested. We have a graph G as an input and S the maximal independent set. We choose the node that have a minimum degree .Then, we add it to S and remove it and its neighbors from V. The following is the most popular greedy algorithm:

Algorithm1 : Maximal Independent Set

```
Input: G = (V, E) a non-oriented graph.
Output: A Maximal Independent Set S ⊆V
Begin
  S ← ∅
While (V≠ ∅) do
       Chose  v as the node of minimum degree in the graph
        S ← S U{ v }
        V ← V–N [v]     //close neighbors of x.
End while.
End.
```

2.3.3 The distributed algorithm of maximal independent set

Many distributed algorithms have been proposed in the literature. The most known is luby's algorithm which is described as follows:

Algorithm 2: Maximal Independent Set(run by every vertex v in parallel)

State: stat ∈ {in,out,act} ,d ∈ {0,1,…,deg(v)}
In each round t=1,2,… do
  if stat= act **then**
    r ← d+1 with probability $\frac{1}{\max\{2d,1\}}$,and r ← 1 otherwise
  else
    if stat = in **then**
      r ← 0
    else
      r ← -1
send message r to all neighbors and receive the message neighbors send
    M ← set of message received
    D ← number of message r′ > 0 received
    If (stat = in □ 0 ∈ M) □ (stat = out □ 0 ∉ M) **then**
      Stat ← act
    else
      if stat = act ∧ 0 ∈ M **then**
        Stat ← out
      else
        if stat = act ∧ max M < r **then**
          Stat ← in
  end
end

2.4 Critical Node Detection Problem

In this section we will give a definition of critical node and critical node detection problem. Then, we talk about variant and application of CNDP.

2.4.1 The critical node

A crucial node in a network is one whose removal would reduce connection and maintain the network's coherence.

2.4.2 Critical node detection problem (CNDP)

The Critical Node Detection Problem (CNDP) is the optimization problem that consists in finding the set of nodes, the deletion of which maximally degrades network connectivity according to some predefined connectivity metrics [21].

2.4.3 Connectivity metrics

The critical nodes can be defined as those the deletion of which disconnects the network according to some predefined connectivity metrics, such as [22]:

- Maximizing the number of connected components.
- Minimizing pairwise connectivity in the network.

- Minimizing the largest component size, etc.

We illustrate this in the graph of Figure2.1:

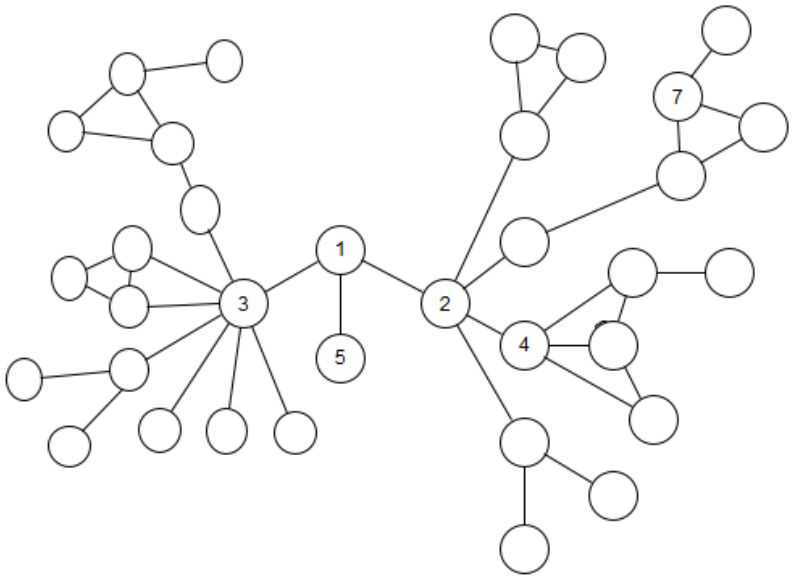

Figure 2.3: A graph to illustrate the optimal solution for different CNDP variants.

- First, we assume that the number of critical nodes to be removed is given, say |S| =k = 1

Optimal solutions for different CNDP variants where a single node is deleted.

The connectivity metric to be satisfied on G[V \ S]	**Optimal solution (graph of Figure2.1)**	**The induced graph G[V \ S]**
maximize the number of components	delete node {3}	generates 7 components
minimize pairwise connectivity	delete node {2}	impairs connectivity to 232 node pairs
minimize component size	delete node {1}	yields a largest component of 16 nodes
maximize the number of smallest components of size <2	delete node {5}	generates four components

Table 2.1: Provides the optimal solution for four different connectivity metrics.

2.4.4 Variants of CNDP

In the general case, CNDP searches for a set of nodes in a graph whose removal minimizes or maximizes a predefined metric. Thus, we have as input a graph G= (V; E) and a connectivity metric named δ, and we output a set of nodes $S \subset V$ such that G [V\S] optimizes the metric. Generally, we have two cases [23]:

- Optimize the metric **δ** so that the number of nodes to be deleted does not exceed k nodes ($|S| \leq k$).
- Minimize the number of nodes to be deleted, such that the metric δ is bounded by a given bound β.

2.4.5 Application of CNDP

Identifying the critical nodes of a network allows us to analyze and understand the structural properties of the structural properties of the network, which simplifies its control, whether the objective is to protect or to protect or destroy it. The two major points that make CNDP an important parameter in the field of in the field of networks are [21]:

- Network applications are typically designed to run in a connected environment, and connected environment, and CNDP is the problem that deals with determining which nodes whose preservation provides such an environment, and whose removal destroys it. Thus, identifying these nodes is very useful for studying network applications before and after design
- CNDP is a double-edged parameter. Indeed, it can be used for offensive or defensive perspectives, depending on the objective of the application to be realized. For example, for an IT network administrator, critical nodes are those that need to be

  protected against virus attacks in order to ensure connectivity in the network (defensive), or they are those that need to be targeted to destroy an adversary network (offensive). Thus, critical nodes can be used in defensive protection and monitoring applications for positive purposes, or in for positive purposes, or in offensive attacks and attempts for negative purposes.

Thus the identification of critical nodes, as an effective means for network analysis, has been applied in many fields for different purposes, including: population vaccination, network vulnerability assessment, social network analysis, telecommunication network security, the study of biological telecommunication networks, the study of biological organisms, etc. In the following, we present the different applications of CNDP classified by domain [22]:

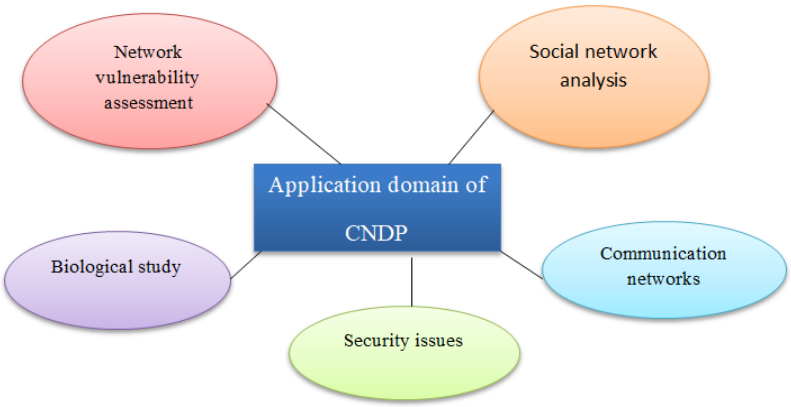

Figure2.4: Application domain of CNDP

- **Biological study:** Biological organisms, such as bacteria and viruses, are sets of interconnected proteins that interact with each other to form a protein-interaction network.
- **Network vulnerability and survivability:** For a long time now, network vulnerability assessment has been of major importance for network risk management. A network could be vulnerable to several potential disruptive events, such as disasters. Generally, network weaknesses are nodes and links, the corruption of which significantly degrades network performance.
- **Communication network study:** In telecommunication networks, identifying critical nodes may be used for both defensive and offensive approaches. Indeed, these nodes are those that must either be destroyed to disable communication capacity in the adversary network or kept intact to avoid any attempt to neutralize the own communication network.
- **Security issues:** Protecting network applications against malicious behaviors is one of the main concerns when designing network applications. A secured application must take into account any network weaknesses that an attacker may exploit.
- **Complex network analysis:** In the literature, the study of complex networks is considered both for analyzing network structure and for studying phenomena and event behavior. Considering a social network, which is an example of complex networks, detection of communities is an example of analyzing the network structure, while examination of contagion spread, product recommendation, trust propagation, information diffusion, etc., are examples of studying phenomena.

2.4.6 Component-Cardinality-Constrained Critical Node Problem (3C-CNP)

➢ **Definition**

3C-CNP seeks to minimize the set of nodes to be deleted while limiting the size of each connected component in the induced graph to a given bound β. Its formulation can be stated as follows[23]:

- Input: an undirected graph G= (V; E) with an integerβ.
- Output: A minimal set of nodes S ⊂V or δ h < β for any component h ϵ G [V\S] .

Algorithm of 3C-CNP

In the literature, a lot of sequential algorithms have been suggested. We explain this algorithm with an example in chapter 3(3.3.2 Programmed Model).The following is the most popular greedy algorithm:

```
Algorithm3 3C_CNP
Input: A graph G = (V, E) , and an integer B ≥ 0
Output: The minimal set of critical nodes
    MIS ← MximalIndependentSet(G);
    RemainNodes ← V/MIS ;
    Critical ← Ø ;
  while (RemainNodes ≠ Ø) do
       i ← argmin_{i∈RemainNodes} (N(V))
      //let i belong to RemainNodes be the node of minimum degree in G
       if ( ∃ c ⊆ G(MIS ∪ {i}): |c| > B ) then
          Critical ← Critical ∪ {i};
          RemainNodes ← RemainNodes / {i};
       else
           MIS ← MIS ∪ {i};
           RemainNodes ← RemainNodes / {i};
       end if
   end while
end.
```

Conclusion

In this chapter, we introduced the basic concepts required for better understanding of graph applications. Then, we talked about distributed system and critical node detection problem in graph.

In the following chapter we will describe our working environment and problem modeling.

Chapter 3: Simulation and Modeling

Introduction

In this chapter, we will talk about simulation and modeling part of our project .We start by giving some basics of simulation. After that, we will present our working environment (Cupcarbon).Finally we talk about problem modeling.

3.1 Simulation Basics

This section is present some basics of simulation and modeling. Firstly, we start by the basic of simulation such as: definition, objective of simulation and its steps. We will end this section by the modeling and model definitions, process and steps of modeling.

3.1.1 Definition of simulation

Simulation is the mechanism of designing, implementing, and investigating a designed prototype of a physical model to check its performance under varying parameters and experimental configurations. This study discusses the numerous advantages such as ease in implementation, economical, easy analysis of real-time scenarios by considering multiple what–if cases. The main objective of this study is to highlight special characteristics of a good simulator such as scalability, completeness, extensibility, and fidelity. A visualisation tool is required in any simulation tool to allow end-users to visualise, monitor, analyse, and debug simulation outcomes. Future work will include the experimental evaluation of different simulators based on the primary metrics such as network throughput and latency, end-to-end delay, packet delivery ratio, and CPU usage under different network scenarios [24].

3.1.2Why simulate?

Generally, the simulation ensures:

- A review of the system's efficiency.
- System control or optimization: calculate the impact of the most important variables to ensure the needed system performance.
- Scenario comparison: contrasting various architectures, tactics, policies, and operating principles.
- Learning: In a virtual world, simulation is frequently utilized for learning. A virtual setting.

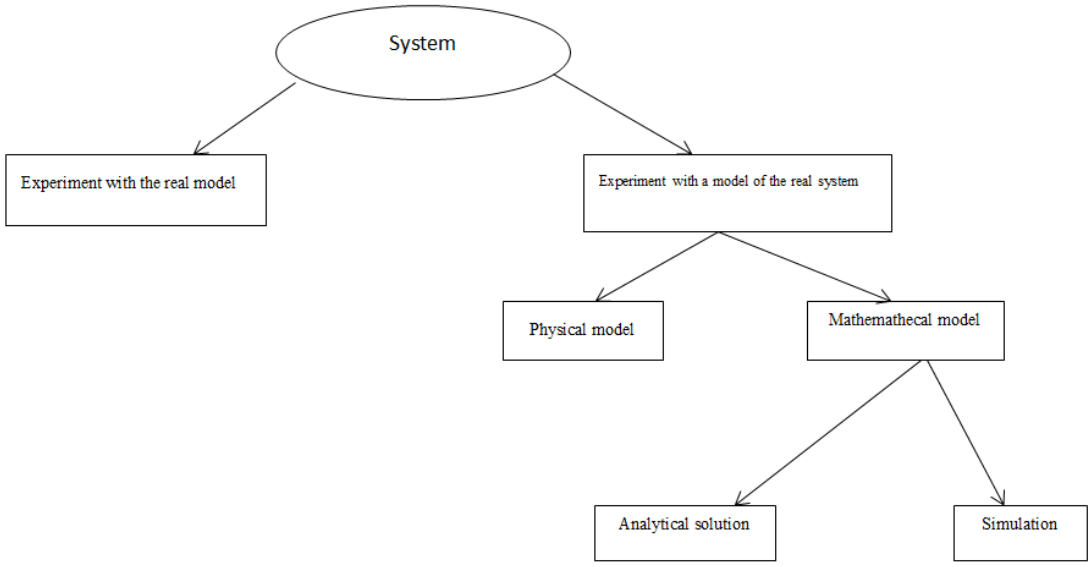

Figure 3.1: Approaches to study a system

3.1.3 Simulation steps

There are mainly five steps involved in designing a simulation model [24].

- Step 1: Decide on the purpose of the simulation. Define a network topology of a model to be simulated.
- Step 2: Setting the values of the simulation parameters. Simulation assumptions are made.
- Step 3: Calibrate and perform simulation based on selected performance metrics.
- Step 4: Visualisation or graphical representation of simulation outcomes.
- Step 5: Analyse the simulation results and select the best alternative.

3.1.4 Definition (Modeling)

Modeling consists of defining the following points: The system, the model, the objective, a criterion of profitability [25].

3.1.5Modeling and simulation process

The steps involved in creating a simulation model can be outlined as follows:

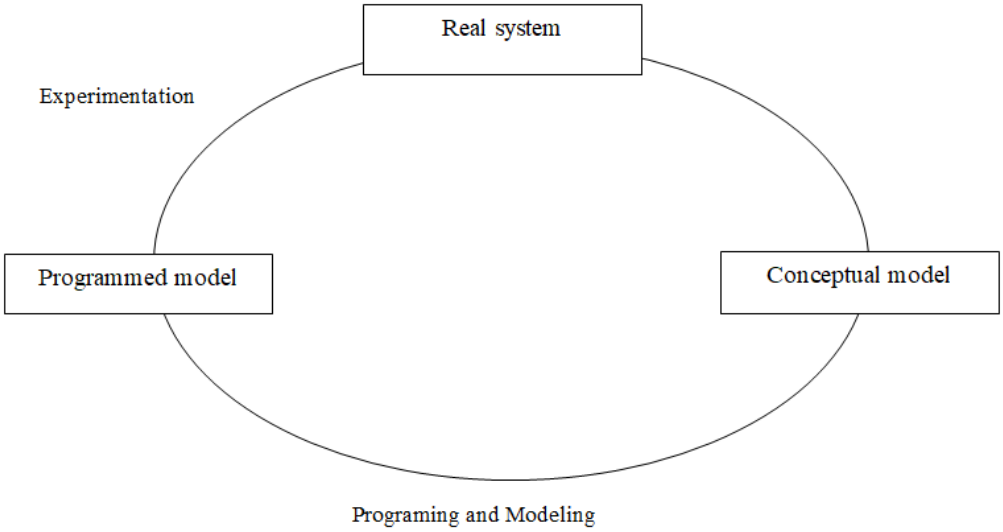

Figure 3.2: Simplified Modeling Process

3.1.6 System modeling steps

The modeling step is an essential phase in the simulation. Different points must be approached [25]:

- Defining the objective of the modeling.
- Define the elements of the system (via the realization of a function, or a process)and system boundaries (inputs, outputs).
- Define the interactions between these elements.
- Define system dynamics.
- Abstraction (choosing the elements of the system relevant to the study).
- Formalization, conceptualization: Mathematical model (the discrete laws and the laws continuous), software model (Simulink, Siman Arena), graphical model (Petri nets).

3.1.7 Definition (model)

A model is a diagram, i.e. a mental (internalized) or figurative (diagrams, mathematical formulas, etc.) description which, for a field of questions, is taken as an abstract representation of a class of phenomena, more or less skilfully taken out of their context by an observer to serve as a support for investigation and/or communication".

A model is therefore a representation of a system (real or imagined) whose purpose is to explain and predict certain aspects of the behavior of this system. This representation is more or less faithful because on the one hand the model must be quite complete in order to be able to answer the various questions that can be asked about the system it represents and on the other hand it must not be too complex. So that it can be easily handled. This immediately implies that there is interest in clearly defining the limits or boundaries of the model that is supposed to represent the system [26].

3.2 Simulation of IoT

This section is devoted to define the simulator what we will be working on. Firstly, we explain why we chose the cupcarbon simulator. After that we give a definition and objective of Cupcarbon. Then, we will present some interface of Cupcarbon and function and message primitives.

3.2.1 Simulator

A simulator is a program or machine that simulates a real-life situation, meaning that it creates a virtual version of it, often for the purpose of instruction or experiment, such as a flight simulator [27].

3.2.2 IoT simulators

Finding a suitable simulation environment for IoT systems is a difficult task. In the literature. There are numerous potential platforms for testing and simulating IoT routing protocols. Various aspects like energy efficiency, resources, decentralized collaboration, fault tolerance, simulation scenarios, global behavior, etc.

The already existing simulators, e.g. NS-2, WSNet, Castalia, TOSSIM, etc, mainly focused on routing protocols. However when it comes to radio channel modeling for smart cities or IoT applications, there interference models are simple and unrealistic. The ***CupCarbon*** platform aims to contribute to the following [1]:

- Give detailed study on WSN deployment with consideration of mobility and the availability of radio spectrum.
- Simulate performances and services of WSN in $2D$ as well as in $3D$ for more realistic environment.
- Give detailed analysis of the feasibility of communication under given environment, reliability of the network and network cost,
- Detects potential interference zones, to enhance the communication quality.
- Gives simulation results of the radio propagation in a real urban environment, more fast and accurate and quickly.
- Provides the visualization of simulation results, to debug and validate any developed algorithm.
- The opening of the source code is not only necessary for embedding or modifying the simulation engine, but also gives significant help in debugging simulation models.

3.2.3 Cupcarbon simulator

Figure 3.3: Cupcarbon simulator

- **Definition and Objective of cupcarbon**

Cupcarbon is a simulator specially designed for WSN networks. Its objective is the design, visualization, and validation of algorithms for environmental monitoring, data collection, etc. It allows creating scenarios with environmental stimulus as fires, gases, and mobile objects within scientific and educational projects. It offers two simulation environments. On the one hand, it enables the design of scenarios with mobility and generation of natural events and on the other, the simulation of discrete events in WSNs [28].

- **Presentation of Cupcarbon**

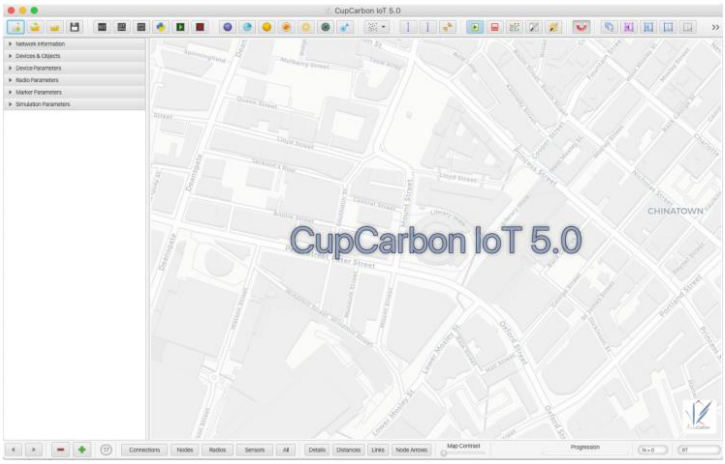

Figure 3.4: Cupcarbon interface

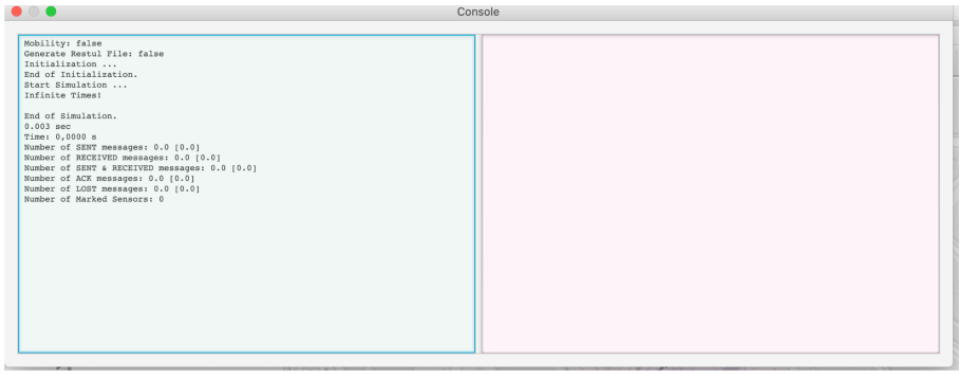

Figure 3.5: The console

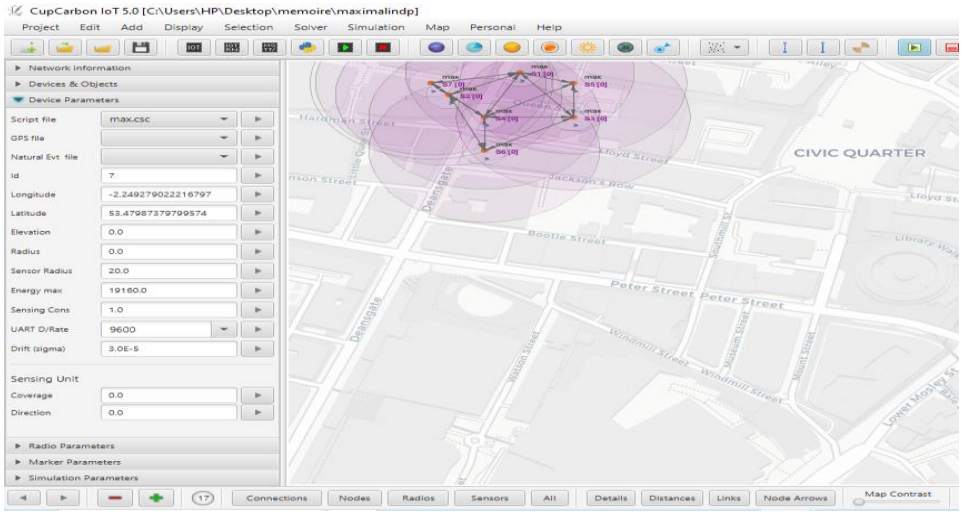

Figure 3.6:Parameters of the selected node

- **User Interface of Cupcarbon**

As Figure 3.7 shows, the Cupcarbon Graphical User Interface (GUI) is composed by six main parts: (1) The map (in the Centre), (2) The menu bar (on the top), (3) The Toolbar (bellow the menu), (4) The parameter menu (on the left), (5) The state bar (at the bottom) and (6) the console (at the bottom).

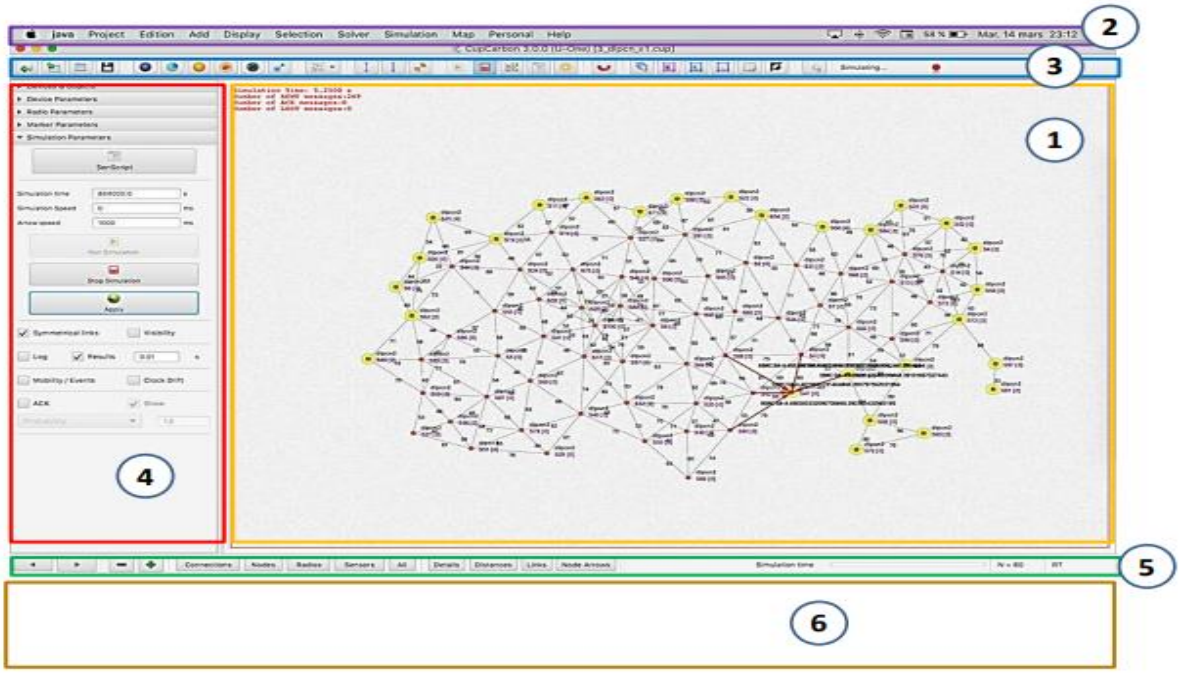

Figure 3.7: User Graphical Interface of Cupcarbon.

Within the selection section of the radio parameters, we can choose between ZigBee,Wifi and Lora. By default, the IEEE 802.15.4 (ZigBee) standard is automatically set for each new added sensor. One of the most interesting competences of this tool is the possibility of deploying the network on different types of maps on which Google Maps stands out. In this way, a more realistic simulation can be achieved. The following devices can be displayed on the different maps:

- Sensor Nodes: nodes in charge of taking the variables of the medium.

Media Sensor Nodes: sensor capable of taking variables from the environment and its sensing unit is directional. It has a form of a cone that can be modified with the SenScript

- Base Station (Sink): It is like a sensor node with the peculiarity that its battery is infinite.
- Mobile: the mobile device can be driven through a route created with markers.
- Markers: these markers can be used for different tasks such as: o Adding sensors can be done randomly delimiting the area on which you want to deploy. O Creating routes. O Adding or drawing buildings [28].

➢ **SenScript**

SenScript is the script used to program sensor nodes of the Cupcarbon simulator. It is intuitive and close to the natural distributed programming language. The nodes can be programmed also with Python for more flexibility and options. The following script shows an example of a SenScript code.

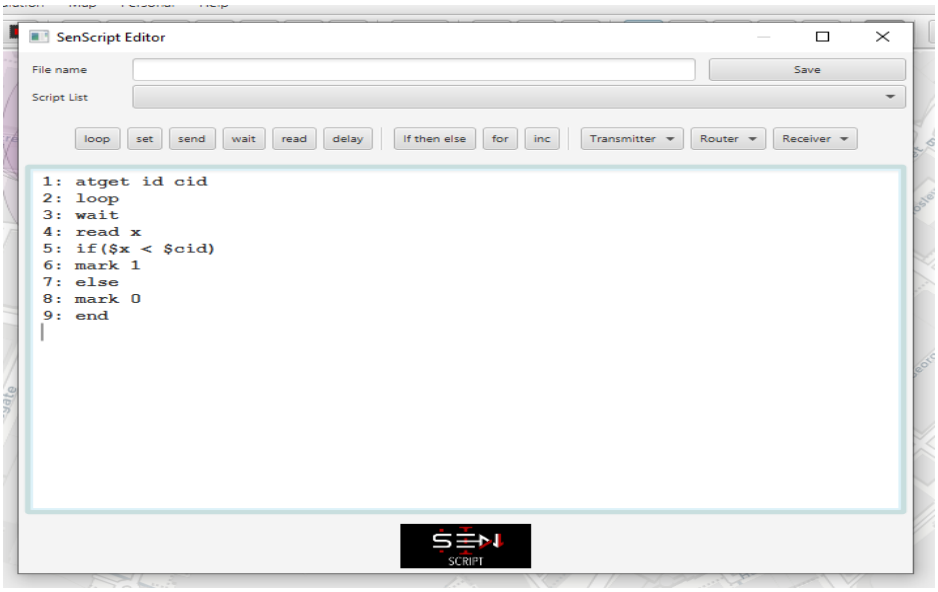

Figure 3.8: SenScript interface

- The command (atget id) of line 1 allows to assign to the variable cid the identifier of the current node.
- The command (loop) allows to start the loop section, where all the code situated after will be executed an infinite number of times.
- The command (wait) will allow to wait for a received message. This command is blocking and the next code will not be executed until a message is received.
- The command (read) of line 4 will assign the message received in the buffer to x.
- In line 5, we test if the received message (an identifier) is lessthan the value of the current identifier cid. If this is the case,line 6 will be executed, and the node will be marked (mark1), otherwise, line 8 will be executed, where the current sensor will be unmarked (mark 0)[29].

3.3 Problem Modeling

We introduce this section by presentation of smart building for simulation.Finally, we will talk about programmed model.

3.3.1 Conceptual model

We will offer a “Smart building” environment considering buildings as nodes contains the firewalls and the arcs it is the link between them.

- **Smart building for simulation and modeling**

Figure 3.9: Smart city

- **Modeling of the smart building by graphs**

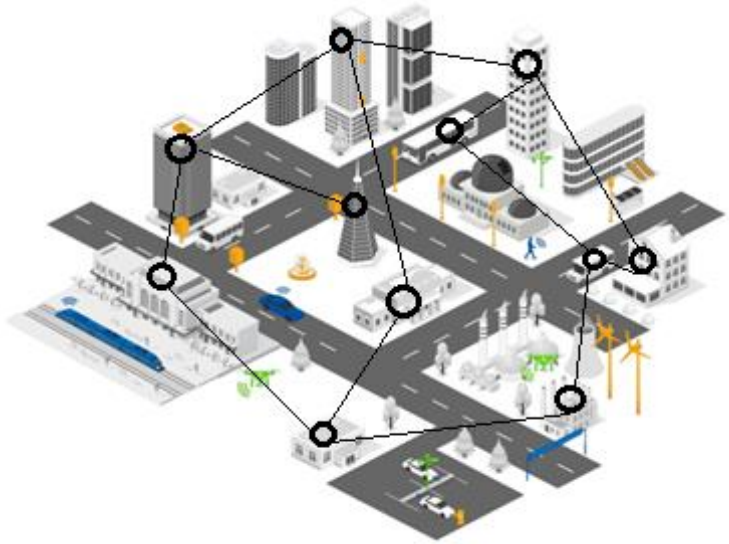

Figure 3.10: Smart building modeling

Smart city ⟶ graphs G(x, u) such that: each vertex represents a building or a house. Each arc represents the relationship between the vertices (cable, wifi ,…). SO the result was a graph G(x, u)/(x = vertex, u = arc).

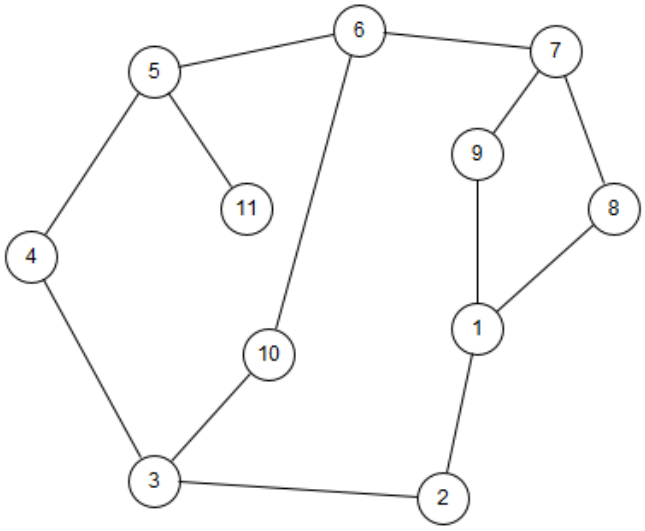

Figure 3.11: Graph result

3.3.2 Programmed Model

- **3C-CNP algorithm analysis**

The algorithm depends on integer B:

- **The case B=1**

Input :G(V,E), V={1,2,3,4,5,6}, E={(1,2),(1,3),(2,3),(2,4),(3,5),(3,7),(4,5),(4,6),(5,6),(5,7)}
Output= MIS, Critical
MIS = {1, 6, 7}
Critical = {2, 3, 4, 5}

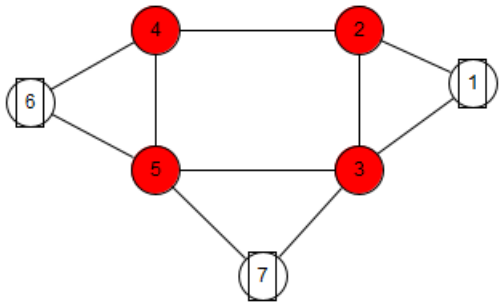

Figure 3.12: Application of3C-CNP for B=1

- **The case B=2**

Input :G(V, E), V={1,2,3,4,5,6}, E={(1,2),(1,3),(2,3),(2,4),(3,5),(3,7),(4,5),(4,6),(5,6),(5,7)}
Output: MIS, Critical
MIS = {1, 2, 6, 7}
Critical = {3, 4, 5}

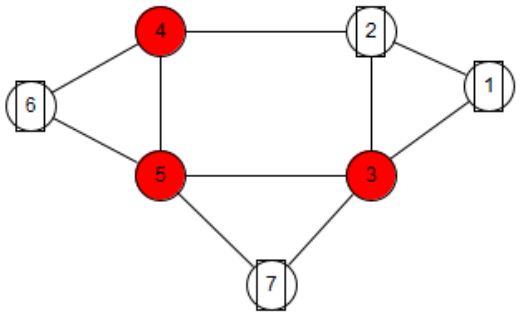

Figure 3.13: Application of 3C-CNP for B=2

- **The case B=3**

Input: G (V, E) ,V={1,2,3,4,5,6}, E={(1,2),(1,3),(2,3),(2,4),(3,5),(3,7),(4,5),(4,6),(5,6),(5,7)}
Output: MIS, Critical

Stage1: MIS = {1, 6, 7}

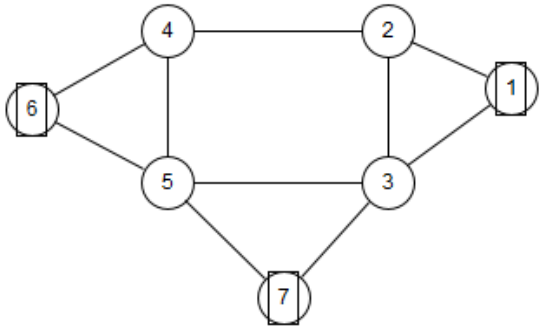

Figure 3.14: Application of 3C-CNP for B=3,stage 1

Stage 2 :

Remain_Nodes = V/MIS = {2, 3, 4, 5}
Critical = Ø

Step1: Remain_Nodes ≠ Ø

i=2: ∀ c ⊆ {{1, 2} {6} {7}}: | {1, 2}|<=2 ^ | {6}|<=2 ^ | {7}|<=2

MIS= MIS ∪ {i} = {1, 6, 7}∪ {2} = {1, 2, 6, 7}

Remain_Nodes= Remain_Nodes/ {2} = {3, 4, 5}

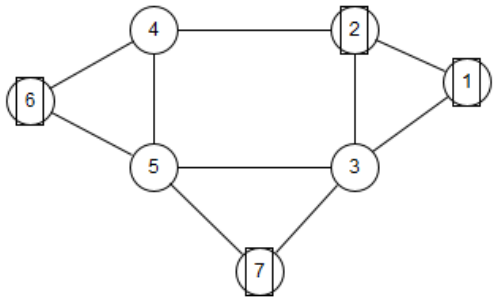

Figure 3.15: Application of 3C-CNP for B=3,stage 2,1

Step2: Remain_Nodes ≠ Ø

i=4 : ∃ c={1,2,4,6} ⊆{{1,2,4,6},{7}}: |c| >=3

Critical = {4}

Remain_Nodes= Remain_Nodes/4= {3, 5}

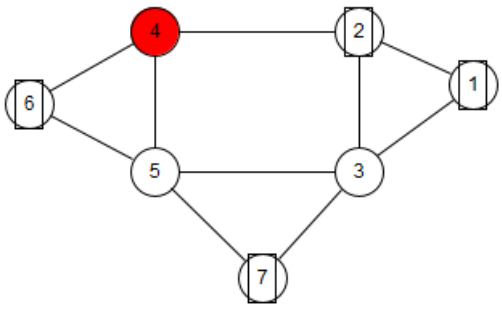

Figure 3.16: Application of 3C-CNP for B=3,stage 2,2

Step3: Remain_Nodes ≠ Ø

i=3: ∃ c⊆ {{1, 2, 3, 7}, {6}}: |c|>=3

Critical = {4, 3}

Remain_Nodes= Remain_Nodes/3= {5}

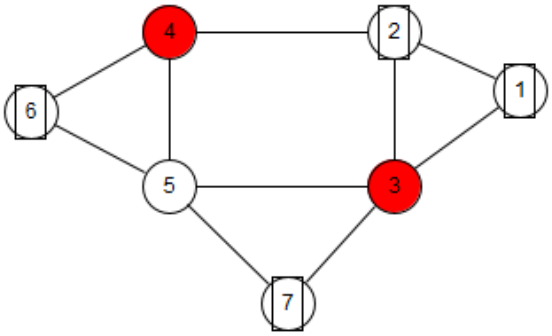

Figure 3.17: Application of 3C-CNP for B=3,stage 2.3

Step4: Remain_Nodes ≠ Ø

i=5: ∄ c ⊆ {{6, 5, 7}, {1, 2}}: |c|<=3

MIS= MIS ∪ {5} = {1, 2, 5, 6,7}

Remain_Nodes= Remain_Nodes/5=Ø

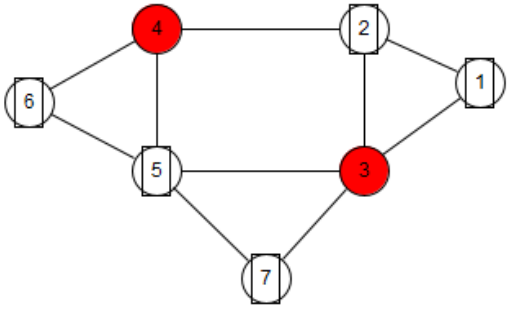

Figure 3.18: Application of3C-CNP for B=3

- **Complexity of the algorithm**

This algorithm need 2 nested loops, the first is to stop the execution. The second is to choose the node.

- **Wait-Before-Starting Model**

Cupcarbon based on this concept of WBS (Wait-Before-Starting) was proposed in this model, assume that:

- The value ***x*** of the node to elect is the minimum one,
- It consists of weighting a value to elect with time:
  - In the case where the value to elect is the maximum, we use the transformation ***x*** = ***c*** − ***x***, where ***c*** is a constant which is greater or equal than ***x***.
  - Then, we can create a relation between the value of any node with a given waiting time unit ***gt***.

- Any node has to wait for a time which is equal to (***t = x * gt***) before starting its program or a part of its program.
- The value of ***gt*** must be sufficiently high to allow the process of selecting the neighbors and decreasing the values of the neighbor of the neighbors to finish.

The pseudo-code of the WBS concept is given by Algorithm [30]:

Algorithm 4 WBS: The pseudo-code of the WBS algorithm

```
Input: x, gt
1: once = false
2: t = x × gt
3: while (true) do
4: r x = read(t)
5: if (r x==null) then
6: send(A, *)
7: stop()
8: end if
9: if (r x==A and once==false) then
10:    once = true
11:    send(A, *)
12:    stop()
13: end if
14: end while
```

- Some function and message primitives

Function	Definition
Send(m,*)	sends the message m in a broadcast
Read()	Waiting for receipt of messages. This function is blocking. If there is no received message anymore, it remains blocked in this instruction
Read(wt)	Waiting for receipt of messages. If there is no received message after wt milliseconds then the execution will continue and go to the next instruction
getCurrentTime()	returns the local time of a node
Stop()	stops the execution of the program

Table 3.1: Functions of the proposed algorithms

Conclusion

In this chapter we given a view of the simulation and modeling technique. Then, we presented the cupcarbon simulator platform, its different interfaces, and its functions .Finally; we presented our conceptual and programming models.

We precised the next chapter for the application of our algorithms on real Algerian cities in order to become a smart city.

Chapter 4: Experimental Results

Introduction

In this chapter, we will describe the city we want to make it smart and why we chose it. Then, we will present our experimental results. Finally, we will compare between our work and the previous works.

4.1 choice of smart city

Given that our study is related to Algerian cities, which number 58 states, there are several large cities (Oran, Alger and Constantine …) that can be applied to, and we chose Ferdjioua because it is located in our state, we know its infrastructure and it is a multi-activity area with many commercial and industrial establishments in addition to public facilities.

The commune of Ferdjioua is located in the northwest of the wilaya of Mila, 37 km west of Mila by the RN79.

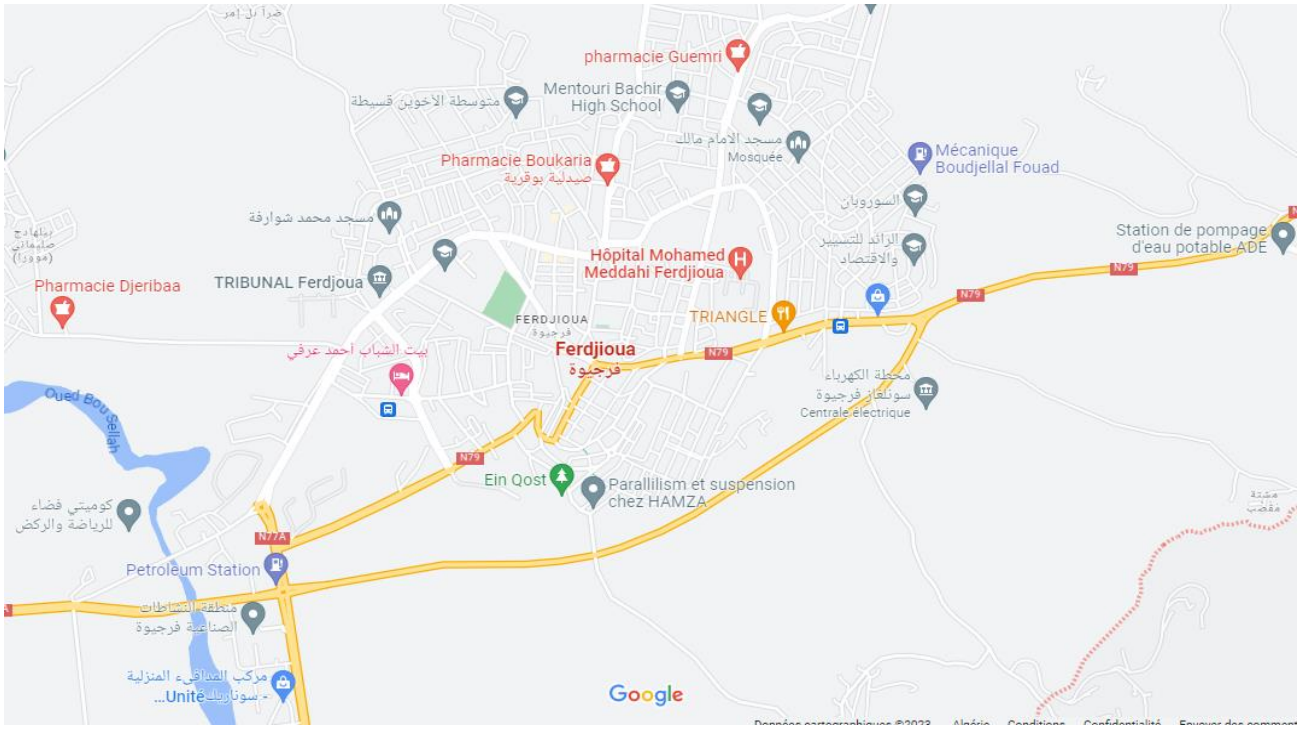

Figure 4.1: Ferdjioua map

4.2 Example of simulation

In this section we will present Mila and Ferdjioua cities before and after simulation. we used an Intel I5 (2.40GHZ) computer with 4 G of RAM under the Windows 10 professional operating system to simulate.

4.2.1 Ferdjioua city

The figure 4.2 and 4.3 represent Ferdjioua city before simulation.

- **Before simulation**

Figure 4.2: Ferdjioua map before simulation (we use Google map)

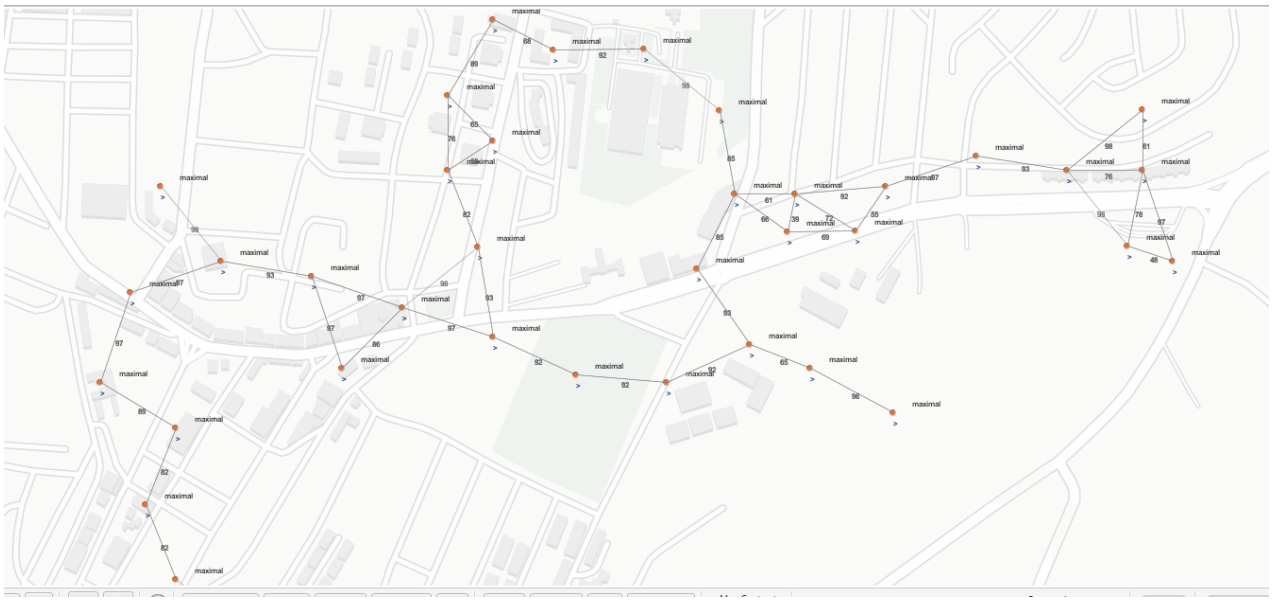

Figure 4.3: Ferdjioua map before simulation (we use OSM light)

- **After simulation**

The figure 4.4 and 4.5 represent Ferdjioua city after simulation.

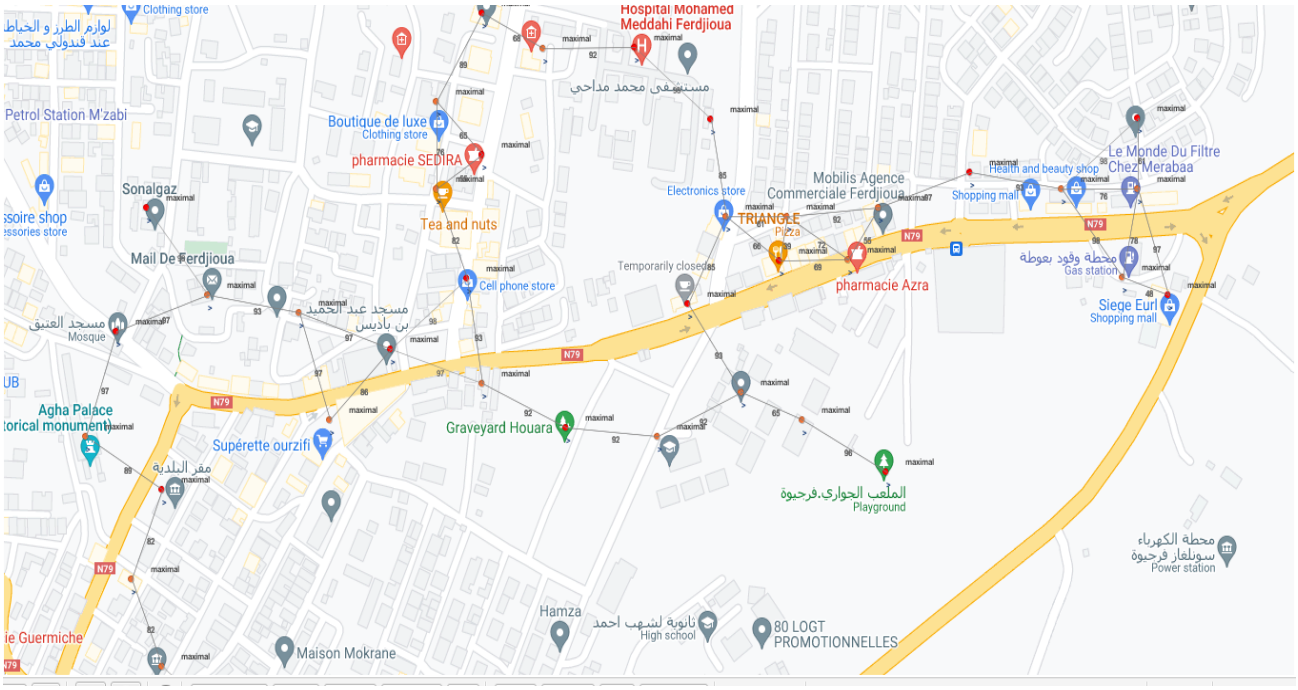

Figure 4.4: Ferdjioua map after simulation (we use Google map)

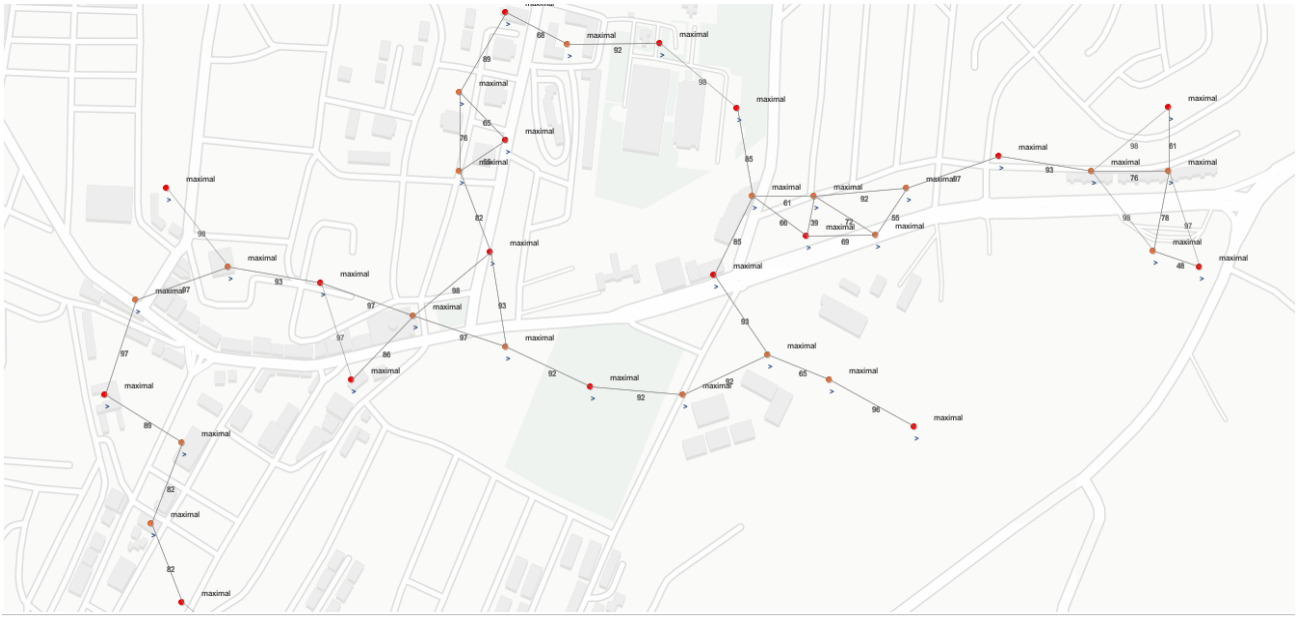

Figure 4.5: Ferdjioua map after simulation (we use OSM light)

4.2.2 Mila city

- **Before simulation**

The figure 4.6 and 4.7 represent mila city before simulation.

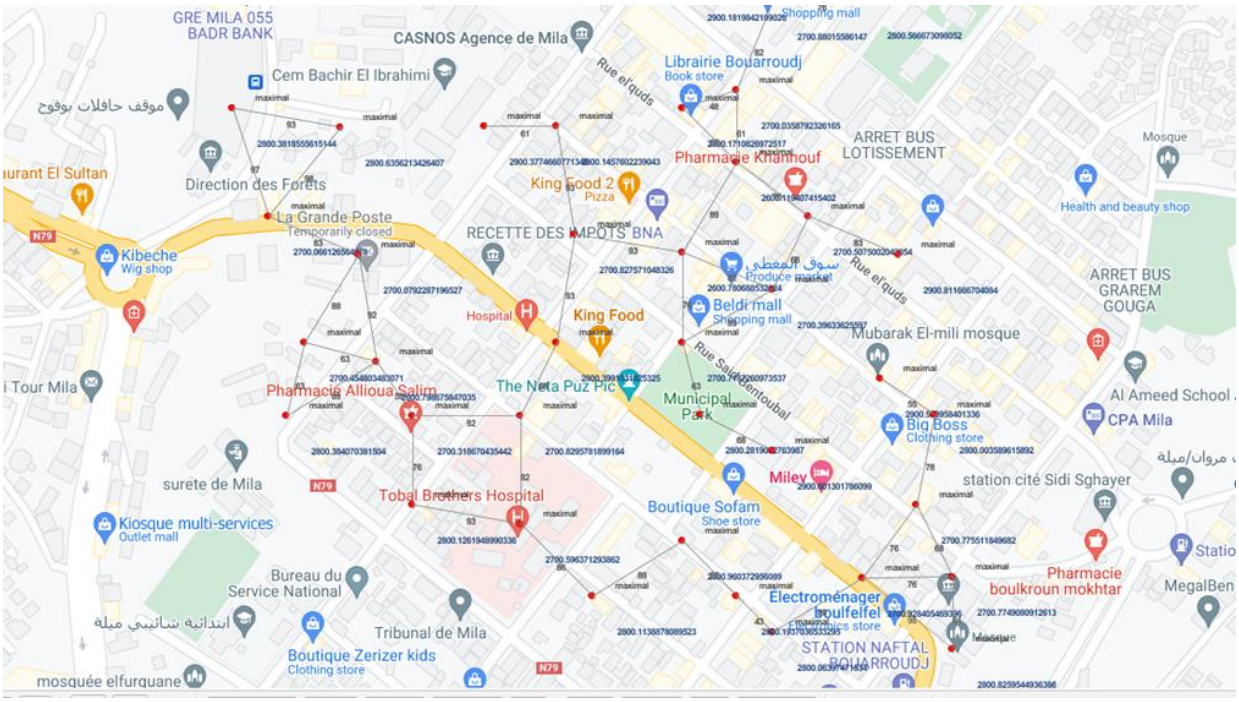

Figure 4.6: Mila map before simulation (we use Google map)

Figure 4.7: Mila map before simulation (we use OSM light)

➢ After simulation

The figure 4.8 and 4.9 represent mila city after simulation.

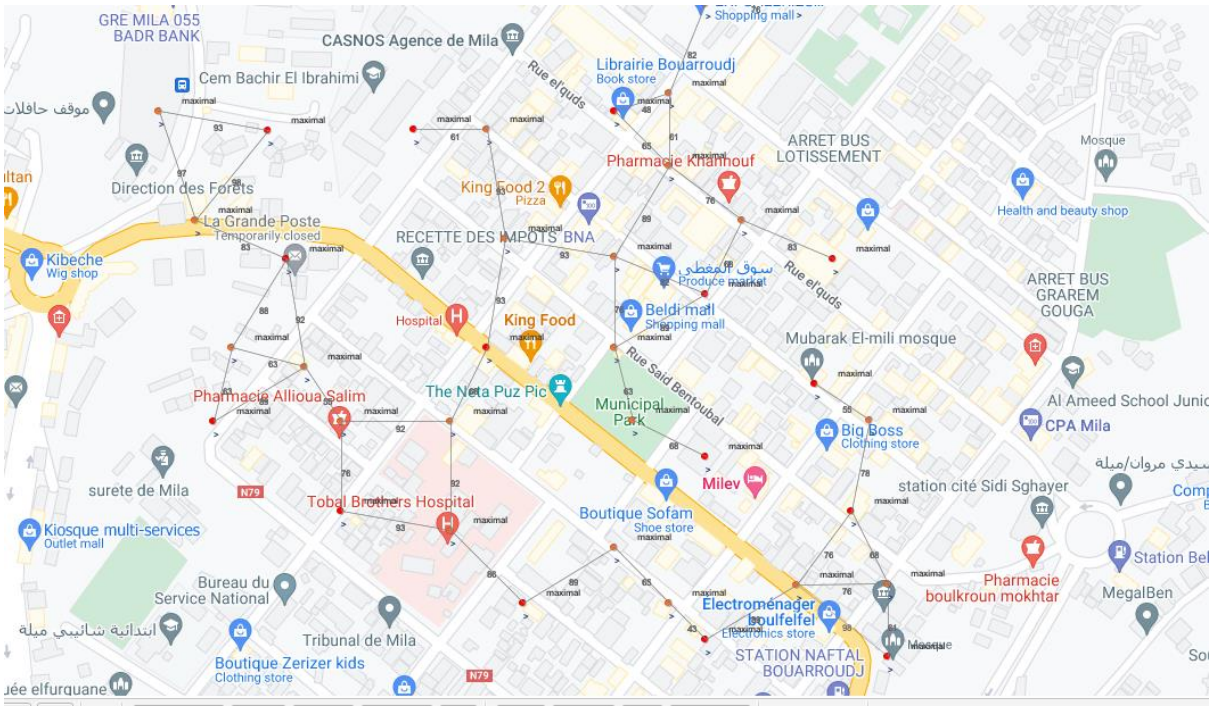

Figure 4.8: Mila map after simulation (we use Google map)

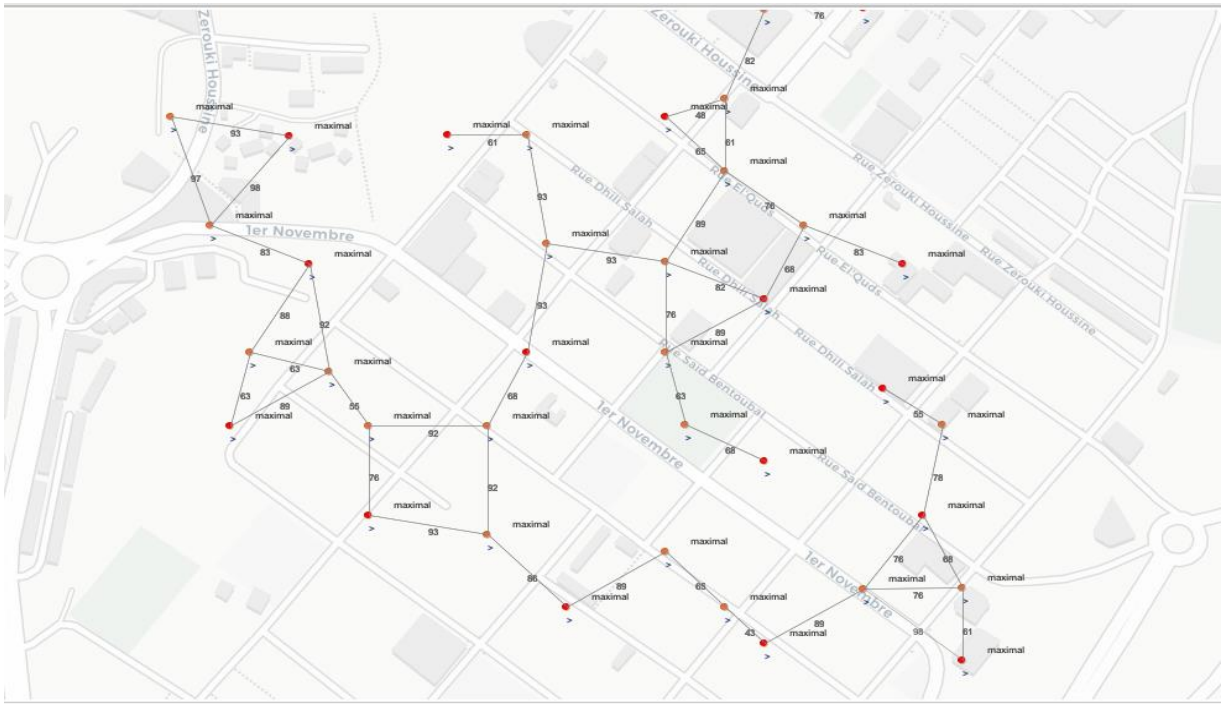

Figure 4.9: Mila map after simulation (we use OSM light)

4.3 Comparison

In this section we will present a comparison between central and distributed algorithm.

4.3.1 Related work

Several distributed algorithms for graph parameters to networks protection have been found in the literature. In IoT networks, a few algorithms have been proposed. In this short survey, we discuss only algorithms witch based on WBS concept: In [31], the authors introduced a greedy sequential algorithm to determine a minimal dominating set in a general undirected connected graph. The algorithm was implemented and applied in a simulated non-real city. In [32], the authors of this paper also proposed a greedy sequential algorithm to find a minimal dominating set in a general undirected connected graph. They further transformed the algorithm into a WBS-distributed algorithm. The resulting algorithm was implemented and applied in a simulated non-real city. In [26], the authors build upon the algorithm proposed in [31]. They transformed the algorithm and implemented it in a real city in Algeria. The specific details and outcomes of the implementation are not provided in the survey.

4.3.2 General comparison (with previous work)

This table (present) gives a comparison between our solution and previous solution

Solution	Work [31]	Work [26]	Work [32]	Our work
Computational algorithm	Central algorithm	Distributed algorithm	Distributed algorithm	Distributed algorithm
Communication between node	Nodes do not communicate with one another via messages.	Synchronous messages are the primary means of communication between nodes.	Synchronous messages are the primary means of communication between nodes.	Synchronous messages are the primary means of communication between nodes.
Execution of algorithm	Step by step, with one node being chosen at a time.	Executed in parallel over a number of nodes.	Executed in parallel over a number of nodes.	Executed in parallel over a number of nodes.
Execution time	The execution time is long.	The execution time is short.	The execution time is superior of Work[26].	The execution time is short.
Security graph parameter	Dominating Nodes Set	Dominating Nodes Set	Dominating Nodes Set	Dominating Nodes Set Maximal Independent set Critical Nodes

Table 4.1: Comparison between our study and previous study

4.3.3 Performance comparison

To assess the performance of our algorithms in relation to the algorithms mentioned above, we have selected execution time as the primary criterion for comparison. This choice is motivated by the influence execution time has on energy conservation for IoT nodes, as well as its impact on connectivity, availability, and security. In order to conduct our evaluations, we have used an undirected connected graph with a maximum of 100 nodes. So, we have generated 10 IoT networks.

Figure 4.10 shows the comparison.

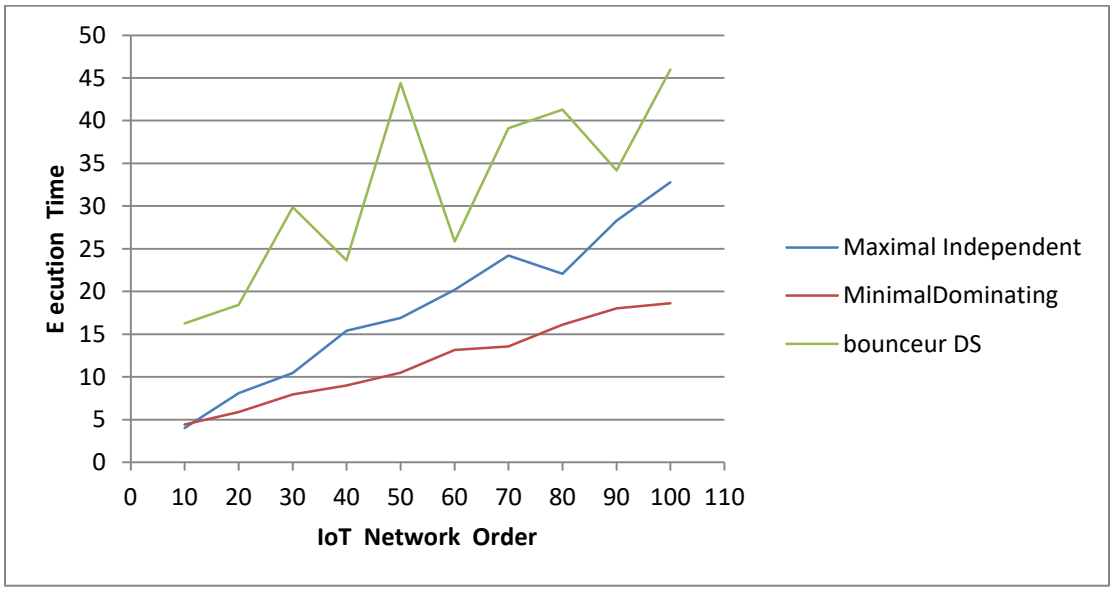

Figure 4.10: Comparison between our MIS algorithm and previous algorithms

4.3.3 Other result

In a graph G= (V, E), and a Maximal Independent Set (MIS):

- The set V- MIS represent a vertex cover, A vertex cover of an undirected graph is a subset of its vertices such that for every edge (u, v) of the graph, either ‘u’ or ‘v’ is in the vertex cover. Although the name is Vertex Cover, the set covers all edges of the given graph. Given an undirected graph, the vertex cover problem is to find minimum size vertex cover [33].

Conclusion

In this final chapter, we concluded by choose of smart cities and presented some interfaces to illustrate the simulation in Algerian cities. Then, we compared the obtained results with results of the previous projects.

General conclusion

Our project consists in proposing and developing a solution that assists network administrators in the design and securing of smart cities based on graph parameters and especially the critical node detection problem in order to reduce the cost of installing different devices and protect the network at the same time. Our solution takes in consideration the system's topology and allows identifying the important key nodes such as: Dominating Nodes, Independent Nodes, Critical Nodes, and Vertex Cover as a new result.

In order to carry out our work, we followed an iterative and incremental approach as follows:

We have introduced the concepts of IoT technologies and their application in smart cities. Furthermore, we have provided a comprehensive overview of diverse security approaches aimed at ensuring the protection of IoT-based environments. Also, we have conducted a study on the problems of MIS and CNDP using both sequential and distributed approaches. After that, we have conducted a thorough search of existing IoT simulators and ultimately selected cupcarbon due to its ability to meet our requirements for simulating our solution.

During the modeling and simulation phase, we represented the smart city as a connected and undirected graph. Subsequently, we implemented the model using Senscript as the programming language for CupCarbon, adhering to the WBS concept. During the experimentation phase, we selected two Algerian cities to create various scenarios. Subsequently, we implemented existing solutions from the literature and programmed them alongside our solution to conduct a comparative analysis.

It is well established that there is no such thing as a perfect work. In order to enhance our approach, it is necessary to incorporate additional variants of security parameters. Furthermore, the implemented algorithms can be further optimized (case B>1). To improve performance, it is essential to include network analysis parameters, such as various centrality measures, in the application.

References

[1] Umber Noreen, Modélisation d'interférence pour Simulateur 3D de Réseaux de Capteurs dédiés aux Villes Intelligentes, thèse de doctorat, université de bretagne occidentale, France.(2018).

[2]https://www.ferrovial.com/en/innovation/technologies/internet-of-things/

[3] https://www.objetconnecte.com/architecture-reseau-iot/

[4]CHALLAL, Yacine. Sécurité de l"Internet des Objets : vers une approche cognitive et Bibliographie 119 systémique. Thèse de doctorat : Technologies de l"Information et des Systèmes. France : Juin 2012, 78 p

[5]https://www.fun-mooc.fr/fr/cours/defis-technologiques-des-villes-intelligentes-participatives/

[6]https://www.connaissancedesenergies.org/fiche-pedagogique/reseau-intelligent-smart-grid

[7]https://wikimemoires.net/2019/09/domaines-d-applications-de-l-iot/

[8]Mr Hidjeb Ali, implémentation d"un protocole d"électiond"un serveur d"authentification dans l'internet des objets, Mémoire de fin de Cycle Master 2 Informatique Professionnel Option: ASR Administration et Sécurité des Réseaux, Université Abderrahmane Mira de Bejaïa .

[9]DOUMI Abdelmoumain, La sécurité des communications dans les réseaux de capteurs sans fil, en vue d"obtention de master académique, 2017/2018

[10]Mezhoud Assia, Mémoire MASTER STIC : Étude comparative des approches de sécurité dans l'IoT, Centre universitaire de Mila, 2021-2022.

[11]M. Lalou, M.A. Tahraoui, H. Kheddouci, Component-cardinality-constrained critical node problem in graphs, Discrete Appl. Math. 210 (2016) 150–163.

[12] t.ly/iF7C

[13] https://www.microsoft.com/en-us/industry/government/resources/smart-cities

[14]https://economictimes.indiatimes.com/definition/graph-theory

[15]Müller, D. Introduction à la théorie des graphes. Commission romande de mathématique (2011).

[16] https://www.tutorialspoint.com/connected-vs-disconnected-graphs

[17] https://www.gcsu.edu/sites/files/page-assets/node-808/attachments/ballardmyer.pdf

[18] Erciyes, K. Distributed graph algorithms for computer networks (2013).

[19] mourad guettiche., parametre de graphes et protection des r ` eseaux. ´ these de doctorat ` en science.,universite de bejaia., 2019.

[20]https://www.confluent.io/learn/distributedsystems/#:~:text=Telephone%20and%20cellular%20networks%20are,distributed%20in%20areas%20called%20cells.

[21] Lalou, M and Autre. The Critical Nodes Detection Problem in Networks.

[22] Lalou, M., Tahraoui, M. A., & Kheddouci, H. The critical node detection problem in networks: A survey. *Computer Science Review*, *28*, 92-117. (2018).

[23] Amel, Z. O. U. A. R. I. Etude et réalisation d'une application réseau à base de nœuds critiques (Doctoral dissertation, Abdelhafid Boussouf University centre-Mila). (2016).

[24]Sharma, R., Vashisht, V., & Singh, U. (2020). Modelling and simulation frameworks for wireless sensor networks: a comparative study. IET Wireless Sensor Systems, 10(5), 181-197.

[25]Abdelkrim, L., &Imene, B. Simulation D'un Système De Production Avec Arena. *UniversiteAbdelhamidIbnBadis-Mostaganem*. (2016).

[26]BercheImane., MoulakmimChaima, Mémoire MASTER STIC: Proposing andimplementing a solution to protect Internet of Things environments, Centre universitaire deMila, 2021-2022.

[27]https://www.dictionary.com/browse/simulator#:~:text=A%20simulator%20is%20a%20program,such%20as%20a%20flight%20simulator.

[28] Lopez-Pavon, C., Sendra, S., & Valenzuela-Valdés, J. F. Evaluation of CupCarbon network simulator for wireless sensor networks. *Network Protocols and Algorithms*, *10*(2), 1-27. (2018).

[29] Bounceur, A., Marc, O., Lounis, M., Soler, J., Clavier, L., Combeau, P., ...& Manzoni, P. (2018, January). Cupcarbon-lab: An iot emulator. In *2018 15th IEEE annual consumer communications & networking conference (CCNC)* (pp. 1-2). IEEE.

[30] Bounceur, A., Bezoui, M., Euler, R., &Lalem, F. A wait-before-starting algorithm for fast, fault-tolerant and low energy leader election in WSNs dedicated to smart-cities and IoT. In 2017 IEEE SENSORS. *IEEE. https://doi. org/10.1109/ICSENS*. (2017).

[31] Bilal, Boulmis, and Lakhal Dounia. *A solution for managing firewalls in the Internet Of Things*. Diss. university center of abdalhafid boussouf-MILA, 2021.

[32] Bezoui, Madani, et al. "A New Distributed Algorithm for Finding Dominating Sets in IoT Networks under Multiple Criteria." *12th International Conference on Multiple Objective Programming and Goal Programming (MOPGP 2017)*. 2017.

[33] https://www.geeksforgeeks.org/introduction-and-approximate-solution-for-vertex-cover-problem/

Annex

1 Traces for experimentation

1.1 Centralized construction

Step0:

G (V, E)
V={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37}
MIS=Ø.
Critical = Ø.
Remain node=V.

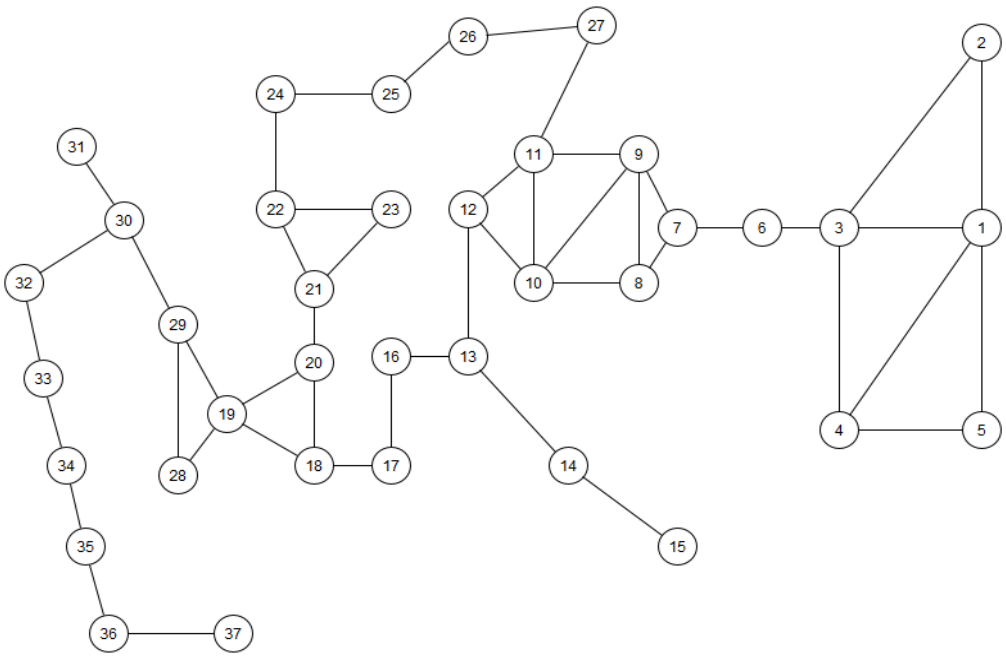

Figure 1: step0

Step1: We choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15}.
Critical = {14}.
Remain_Nodes=V/ {15, 14}.

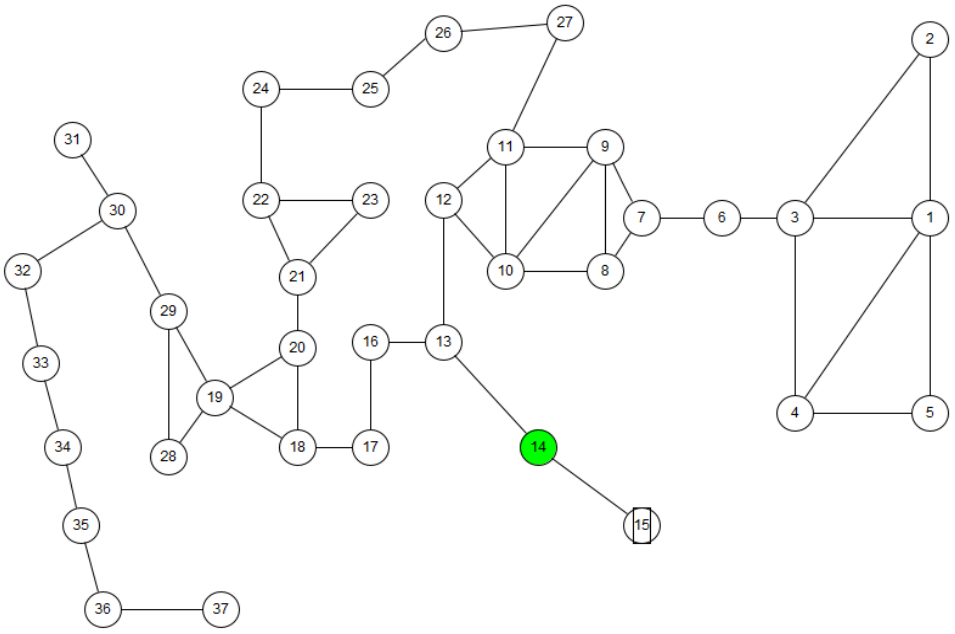

Figure 2: step1

Step2: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 31}.
Critical = {14, 30}.
Remain_Nodes=V/ {15, 14, 31, 30}.

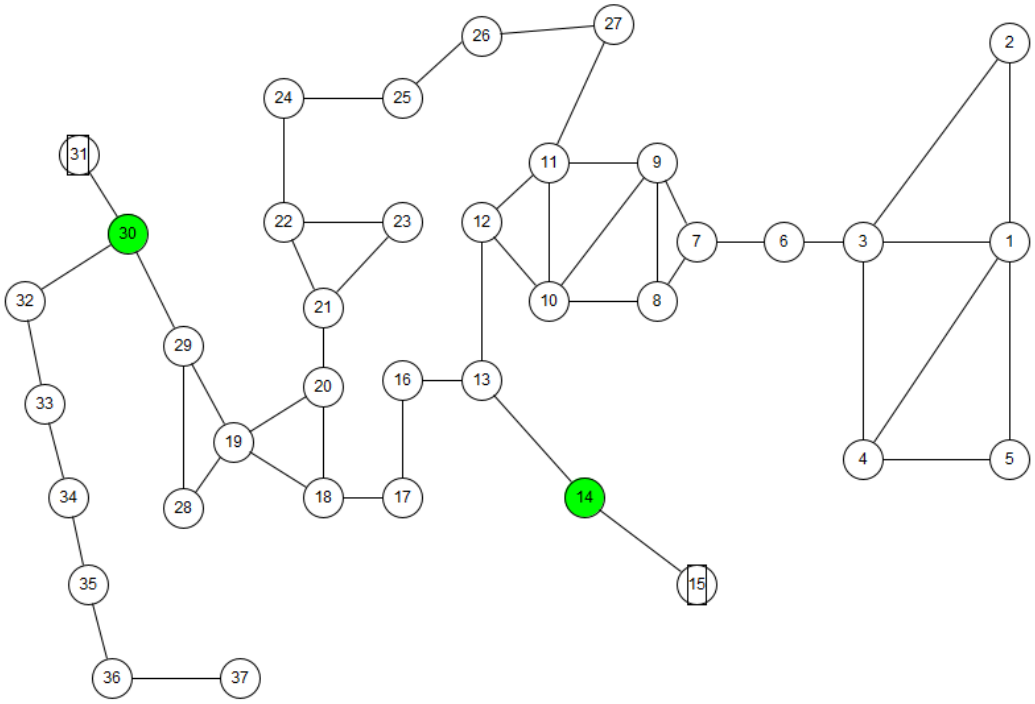

Figure 3: step2

Step3: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 31, 32}.
Critical = {14, 30, 33}.
Remain_Nodes=V/ {15, 14, 31, 30, 32, 33}.

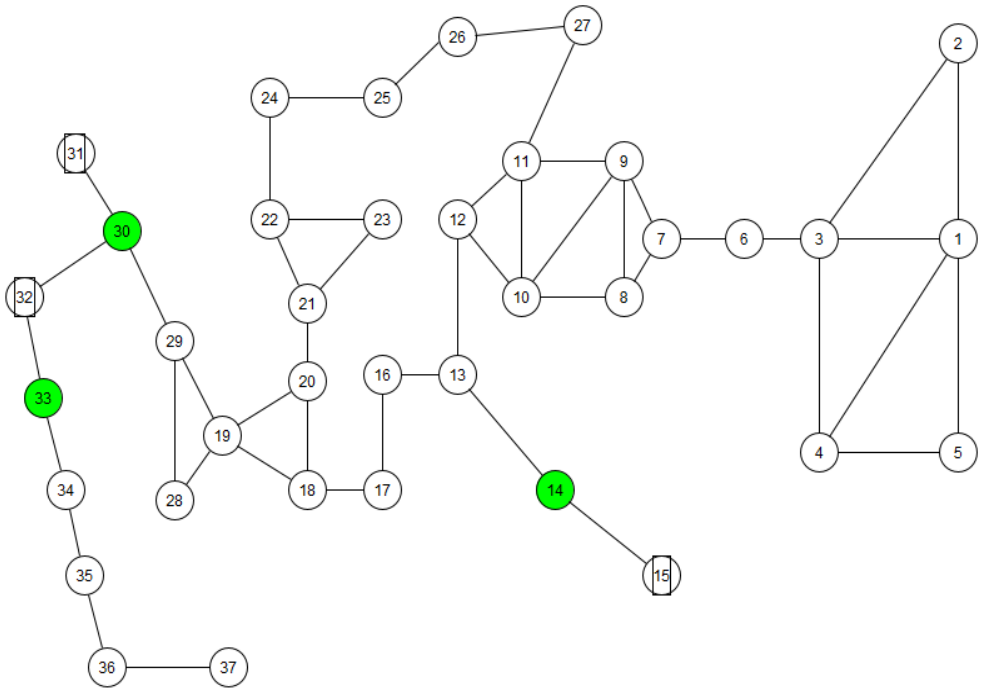

Figure 4: step3

Step4: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 31, 32, 34}.
Critical = {14, 30, 33, 35}.
Remain_Nodes=V/ {15, 14, 31, 30, 32, 33, 34, 35}.

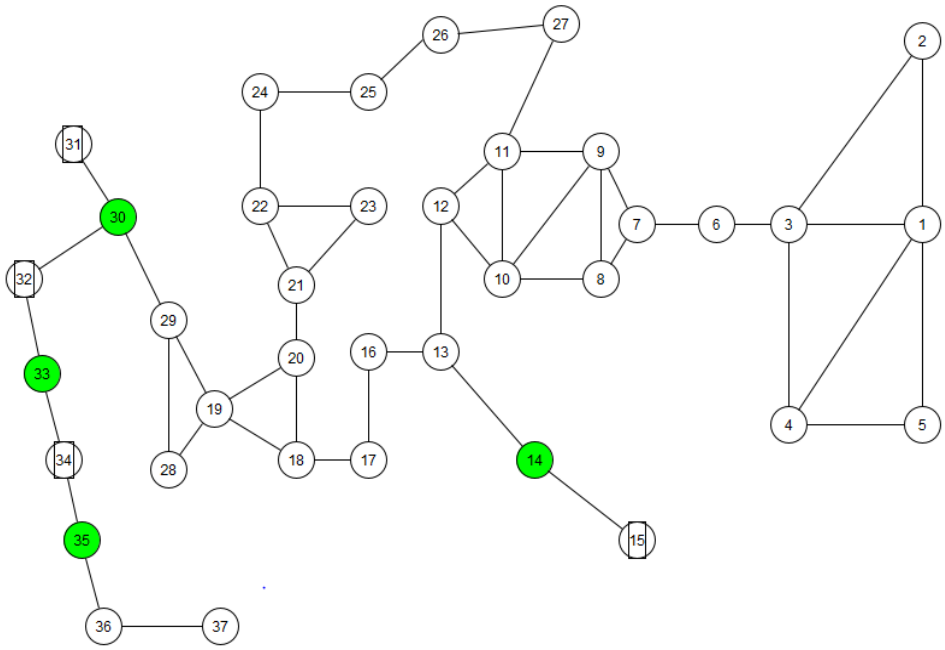

Figure 5: step4

Step5: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 31, 32, 34, 36}.
Critical = {14, 30, 33, 35, 37}.
Remain_Nodes=V/{15,14,31,30,32,33,34,35,36,37}.

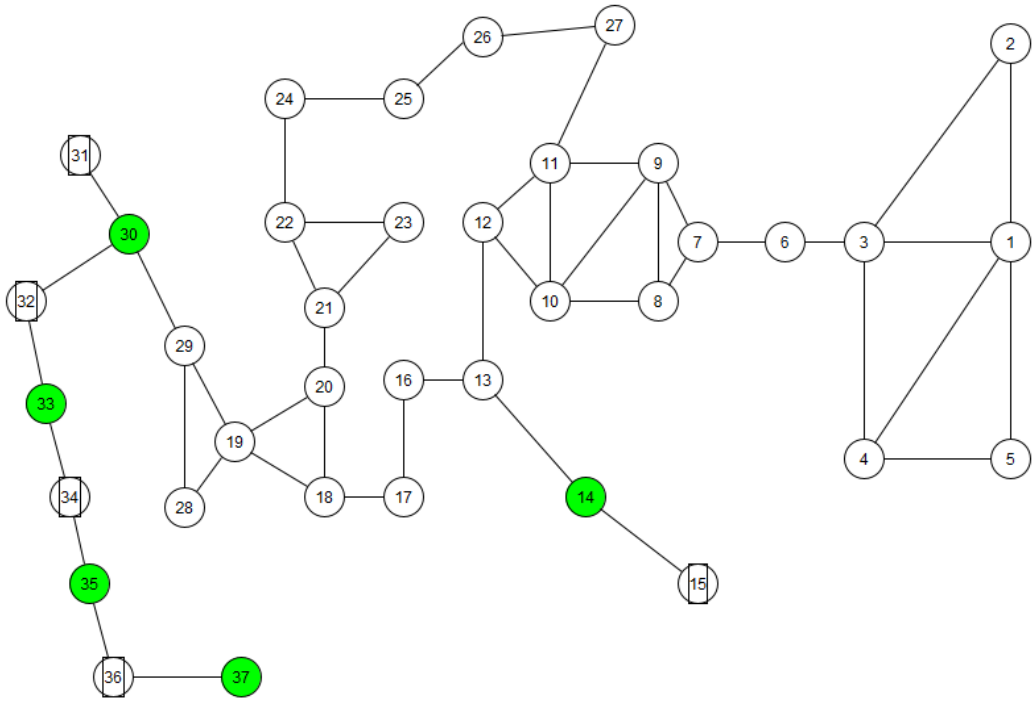

Figure 6: step5

Step6: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 31, 32, 34, 36, 2}.
Critical = {14, 30, 33, 35, 37, 1, 3}.
Remain_Nodes=V/{15,14,31,30,32,33,34,35,36,37,2,1,3}.

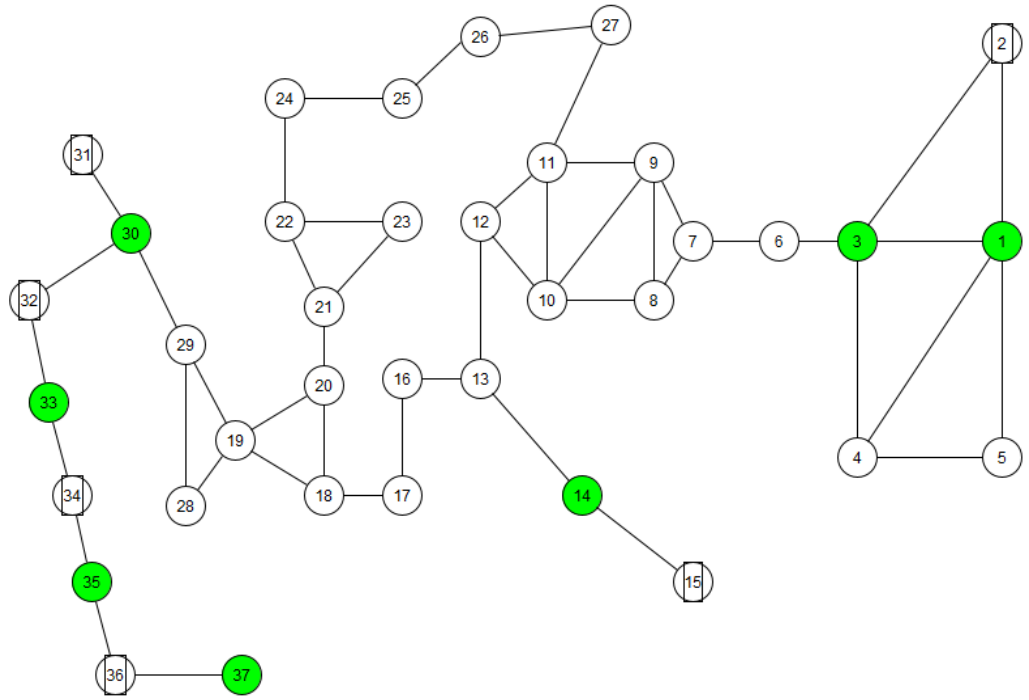

Figure 7: step6

Step7: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 31, 32, 34, 36, 2, 4}.
Critical = {14, 30, 33, 35, 37, 1, 3, 5}.
Remain_Nodes=V/{15,14,31,30,32,33,34,35,36,37,2,1,3,4,5}.

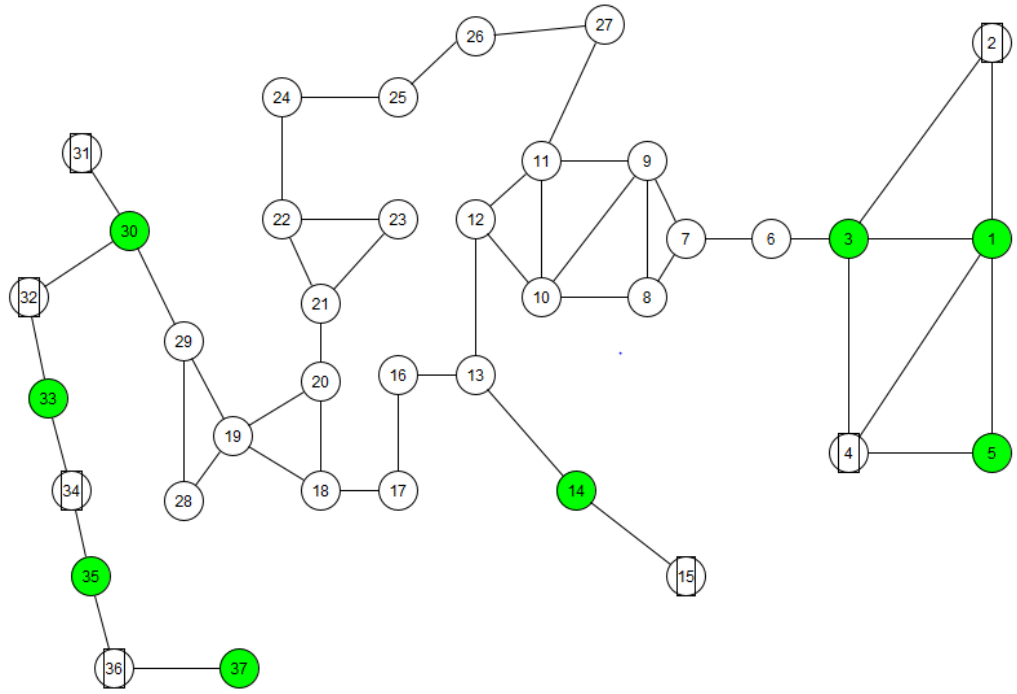

Figure 8: step7

Step8: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 31, 32, 34, 36, 2, 4, 6}.
Critical = {14, 30, 33, 35, 37, 1, 3, 5, 7}.
Remain_Nodes=V/{15,14,31,30,32,33,34,35,36,37,2,1,3,4,5,6,7}.

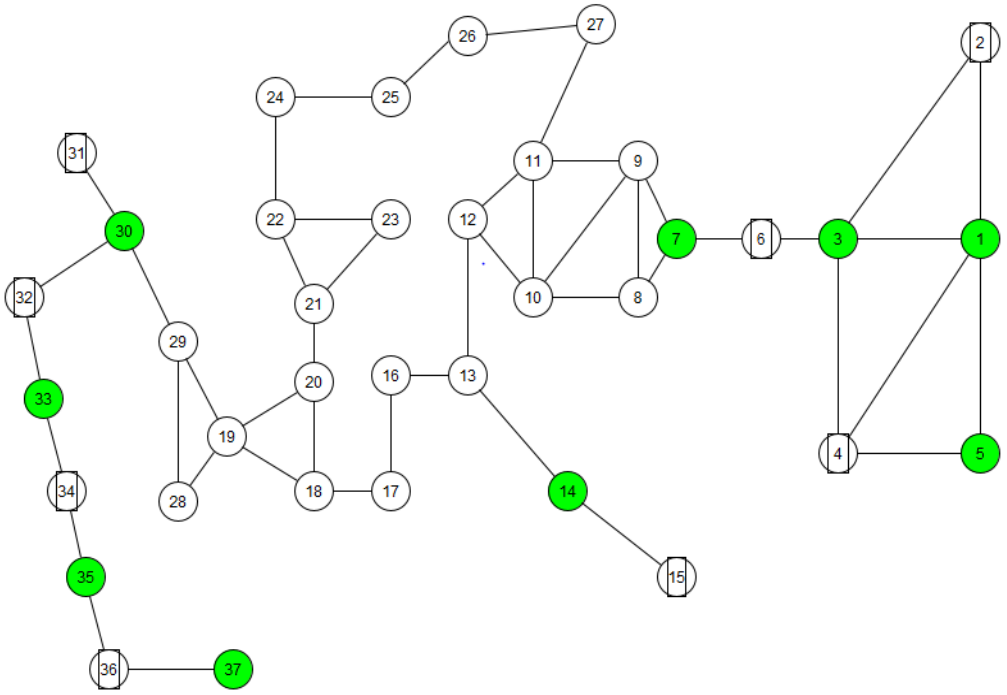

Figure 9: step8

Step9: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 31, 32, 34, 36, 2, 4, 6, 8}.
Critical = {14,30,33,35,37,1,3,5,7,9,10}.
Remain_Nodes =V/{15,14,31,30,32,33,34,35,36,37,2,1,3,4,5,6,7,8,9,10}.

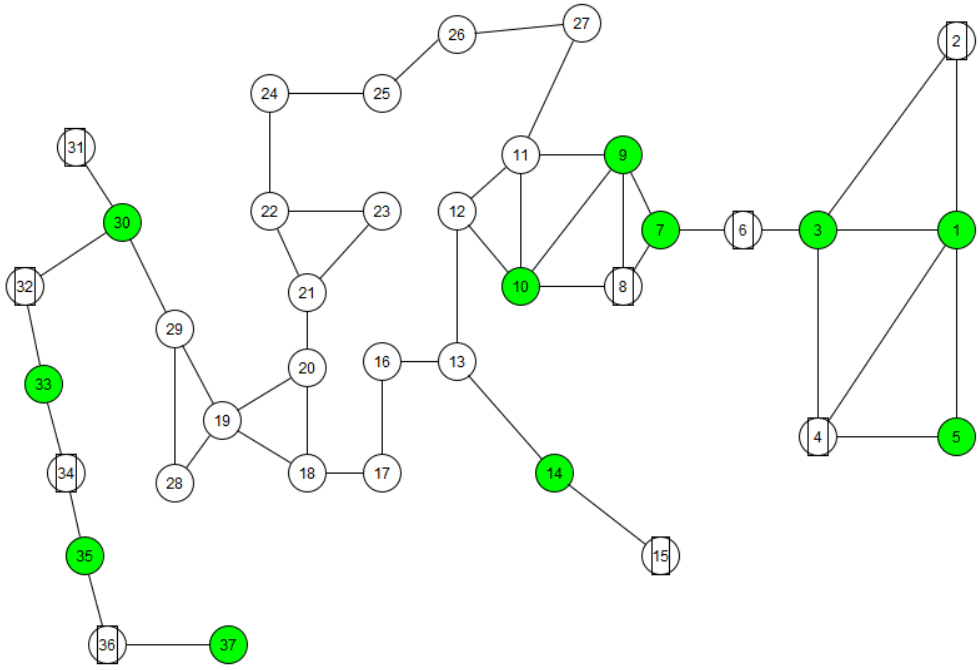

Figure 10: step9

Step10: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 31, 32, 34, 36, 2, 4, 6, 8, 11}.
Critical = {14,30,33,35,37,1,3,5,7,9,10,12,13}.
Remain_Nodes =V/{15,14,31,30,32,33,34,35,36,37,2,1,3,4,5,6,7,8,9,10,11,12,13}.

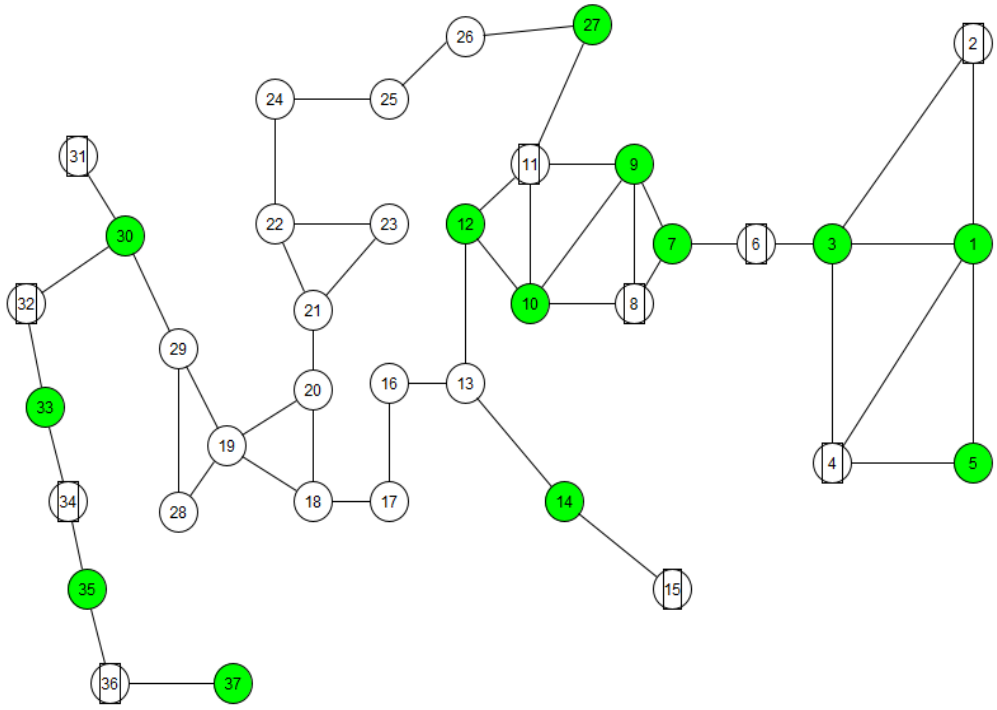

Figure11: step10

Step11: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,31,32,34,36,2,4,6,8,11,13}.
Critical = {14,30,33,35,37,1,3,5,7,9,10,12,27,16}.
Remain_Nodes =V/{15,14,31,30,32,33,34,35,36,37,2,1,3,4,5,6,7,8,9,10,11,12,27,13,16}.

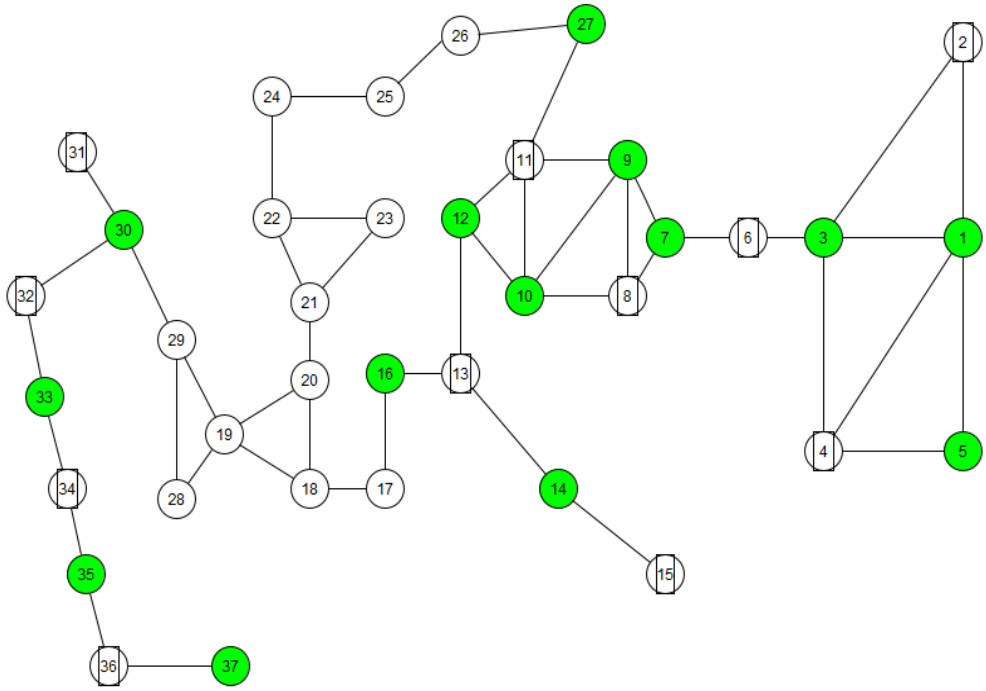

Figure 12: step11

Step12: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,31,32,34,36,2,4,6,8,11,13,17}.
Critical = {14,30,33,35,37,1,3,5,7,9,10,12,27,16,18}.
Remain_Nodes =V/{15,14,31,30,32,33,34,35,36,37,2,1,3,4,5,6,7,8,9,10,11,12,27,13,16,17,18}.

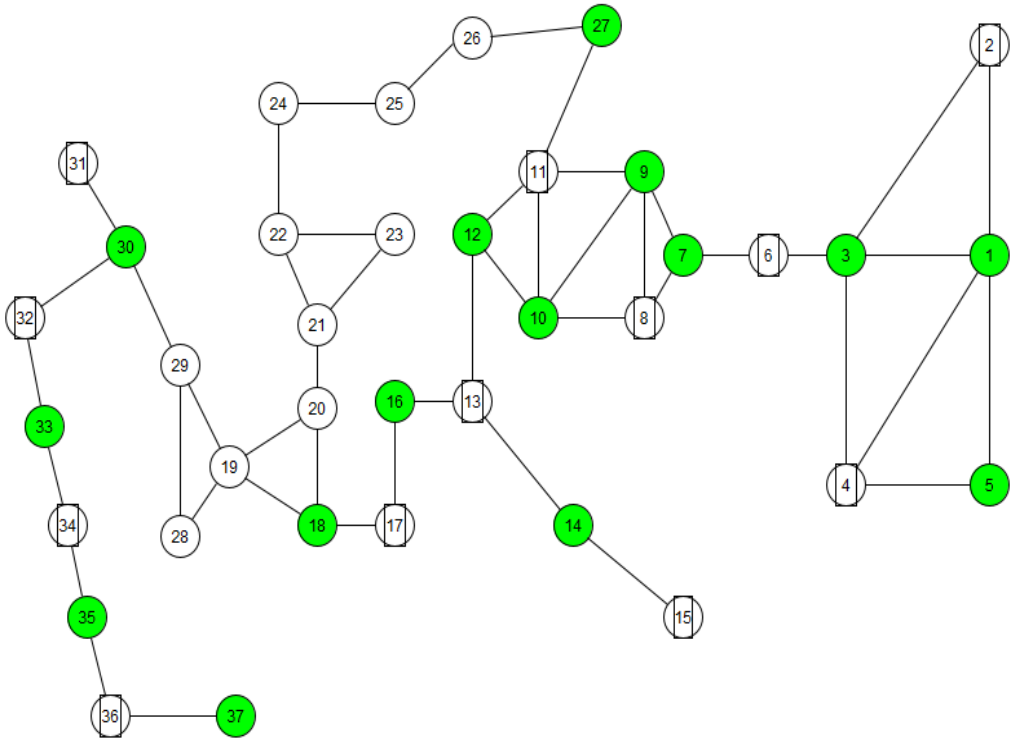

Figure 13: step12

Step13: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,31,32,34,36,2,4,6,8,11,13,17,26}.
Critical = {14,30,33,35,37,1,3,5,7,9,10,12,27,16,18,25}.
Remain_Nodes ={19, 20, 21, 22, 23, 24, 28,29}.

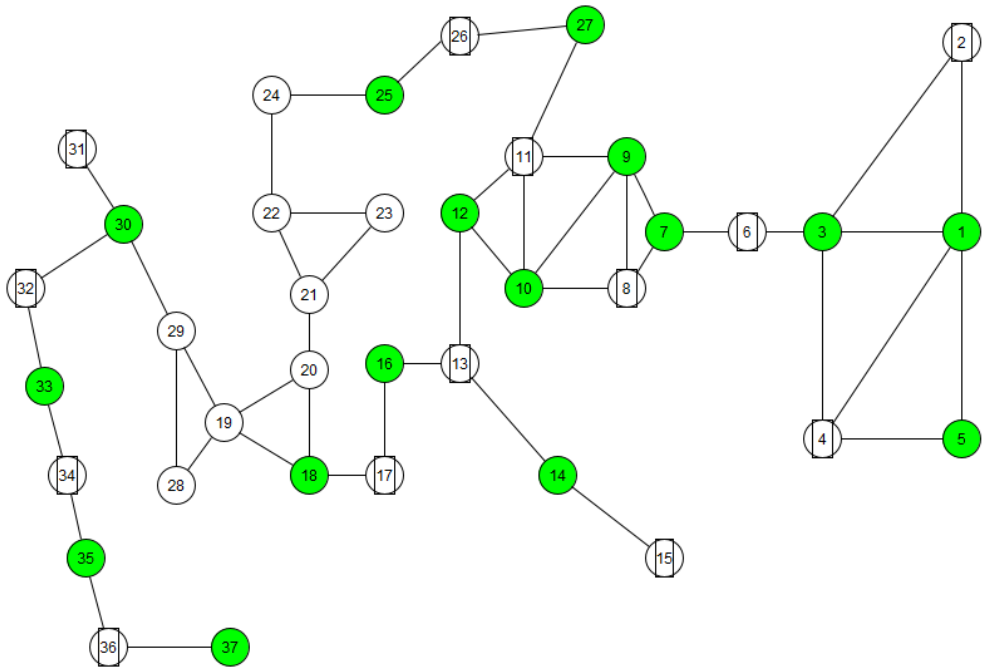

Figure 14: step13

Step14: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,31,32,34,36,2,4,6,8,11,13,17,26,24}.
Critical = {14,30,33,35,37,1,3,5,7,9,10,12,27,16,18,25,22}.
Remain_Nodes ={19, 20, 21, 23, 28, 29}.

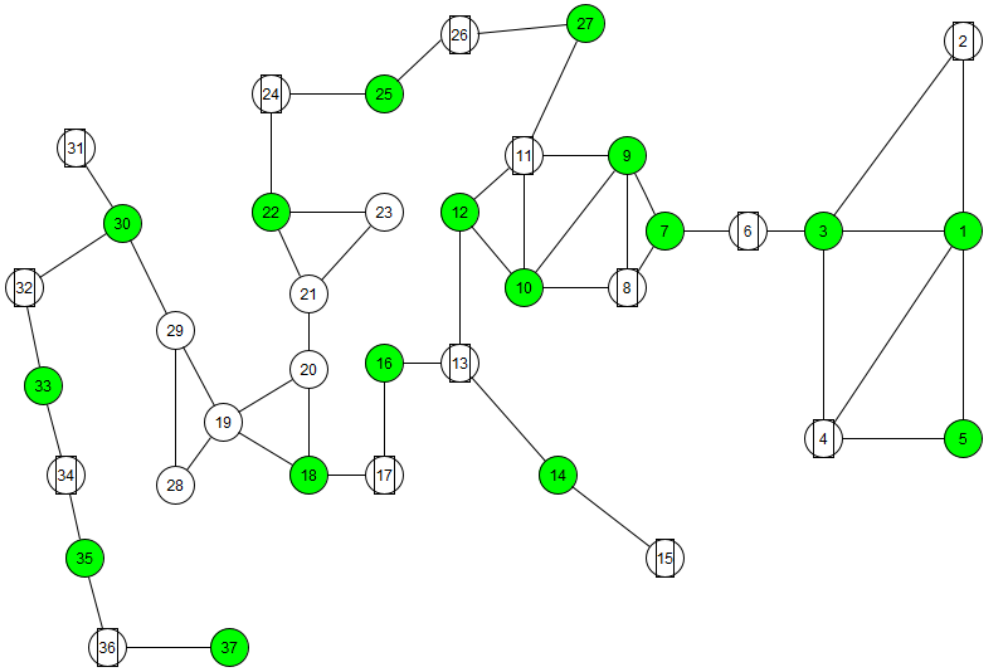

Figure 15: step14

Step15: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,31,32,34,36,2,4,6,8,11,13,17,26,24,23}.
Critical = {14,30,33,35,37,1,3,5,7,9,10,12,27,16,18,25,22,21}.
Remain_Nodes ={19,20,28,29}.

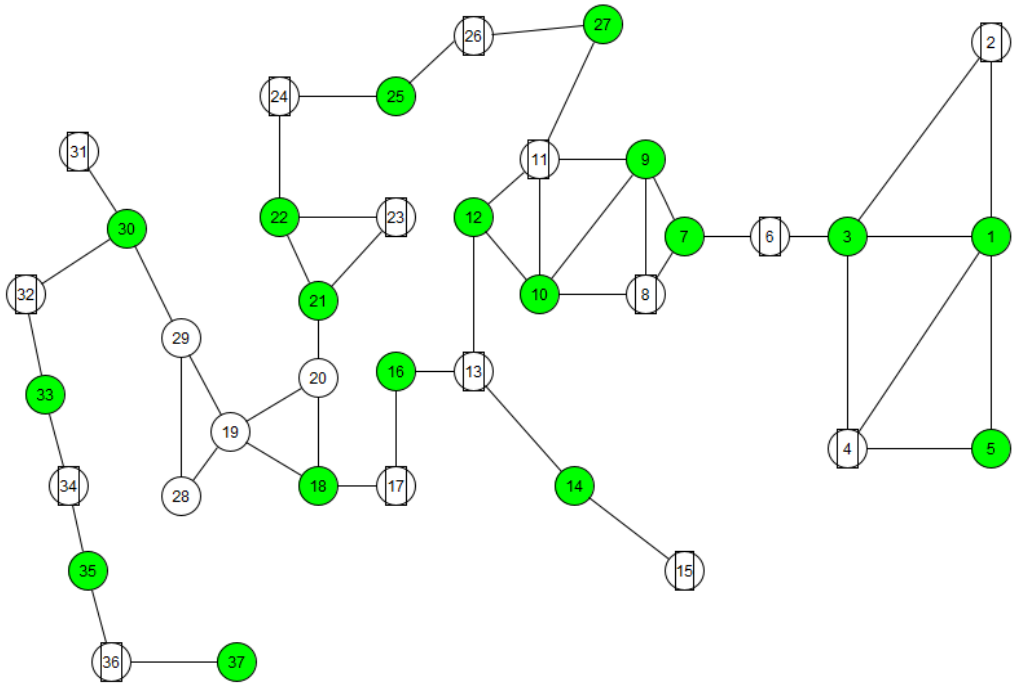

Figure 16: step15

Step16: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,31,32,34,36,2,4,6,8,11,13,17,26,24,23,20}.
Critical = {14,30,33,35,37,1,3,5,7,9,10,12,27,16,18,25,22,21,19}.
Remain_Nodes ={28, 29}.

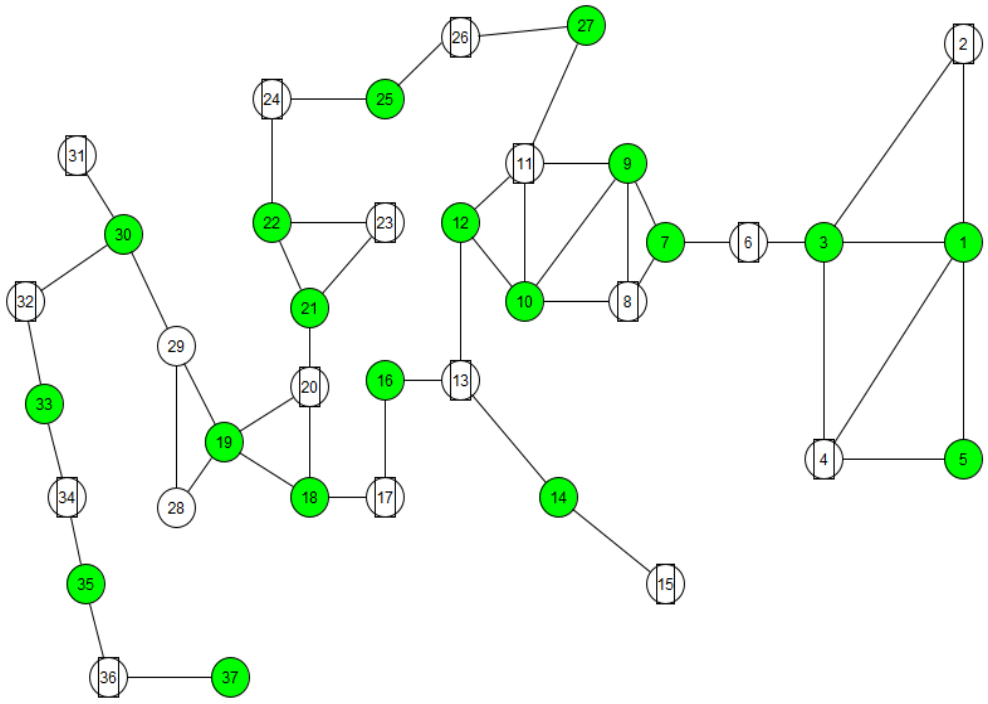

Figure 17: step16

Step17: We do the same work on the new graph we choose one node that have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,31,32,34,36,2,4,6,8,11,13,17,26,24,23,20,28}.
Critical = {14,30,33,35,37,1,3,5,7,9,10,12,27,16,18,25,22,21,19,29}.
Remain_Nodes= Ø.

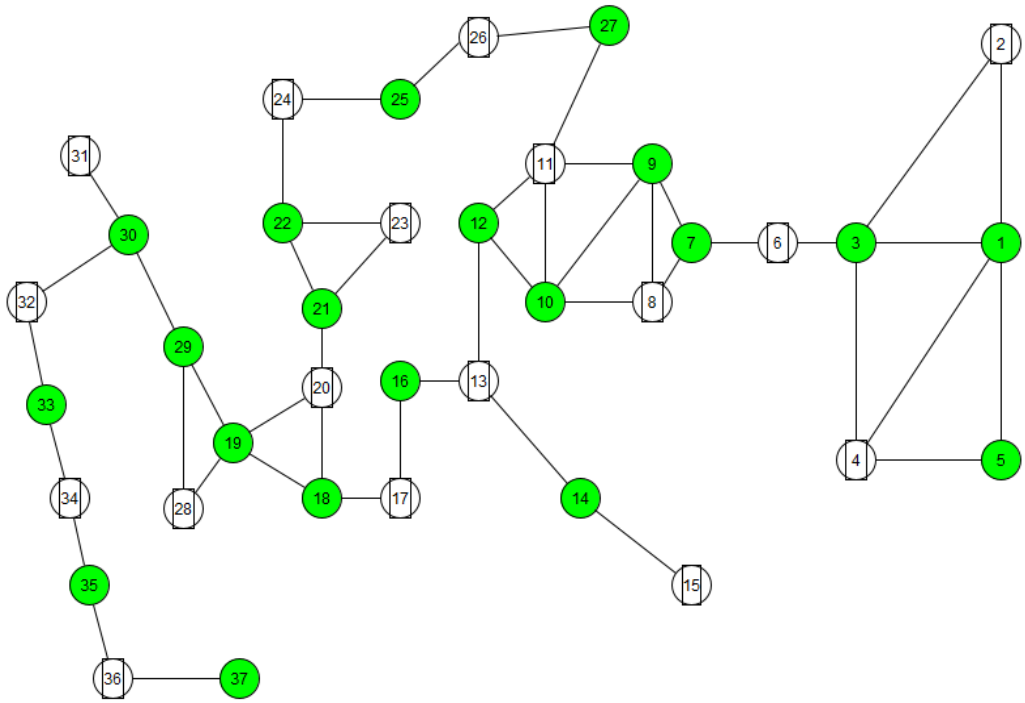

Figure 18: step17

1.2 Distributed construction

Step0:

G (V, E).
V={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37}.
MIS= Ø.
Critical = Ø.
Remain_Nodes =V.

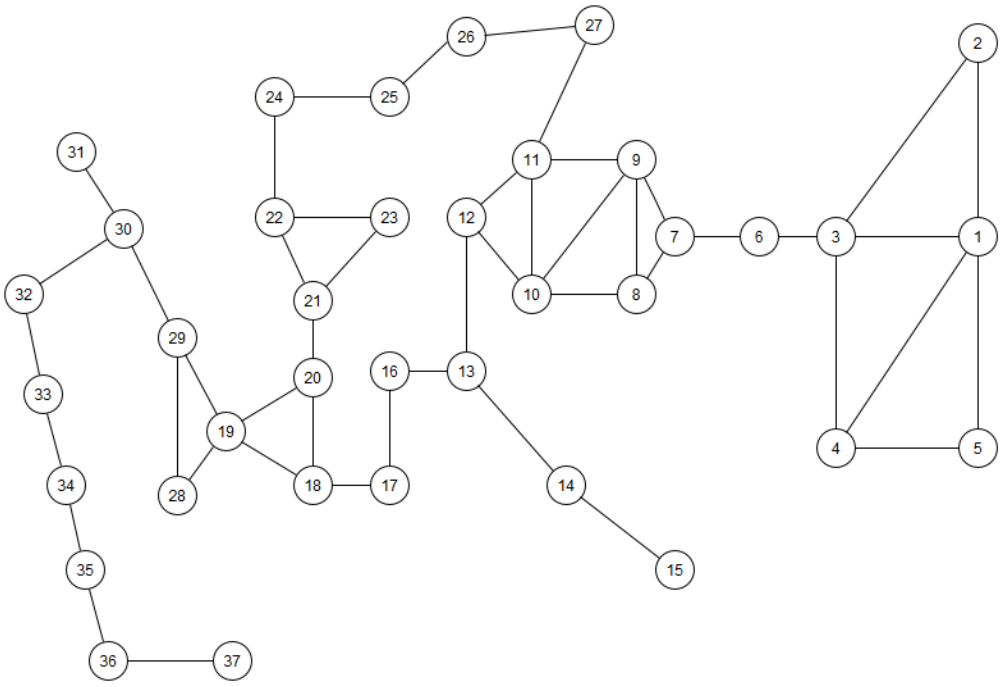

Figure 19: step 0

Step1: We choose the nodes those have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 37, 31}.
Critical = {14, 36, 30}.
Remain_Nodes=V/ {15, 14, 36, 37, 31, 30}.

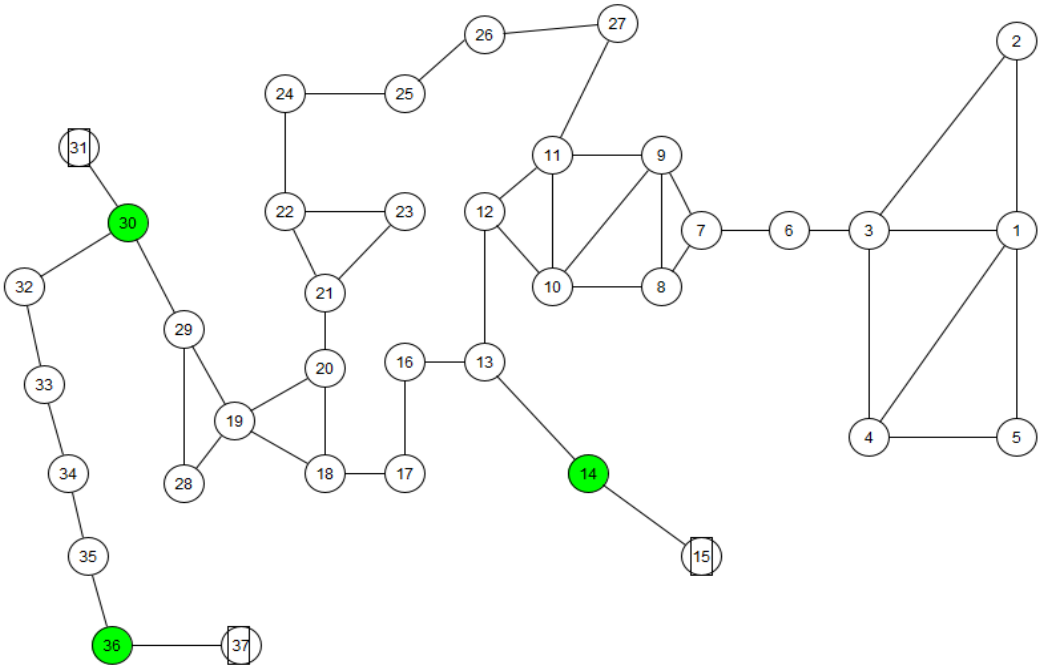

Figure 20: step1

Step2: We choose the nodes those have the minimum degree in the new graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15, 37, 31, 35, 32}.
Critical = {14, 36, 30, 34, 33}.
Remain_Nodes=V/ {15, 14, 36, 37, 31, 30}.

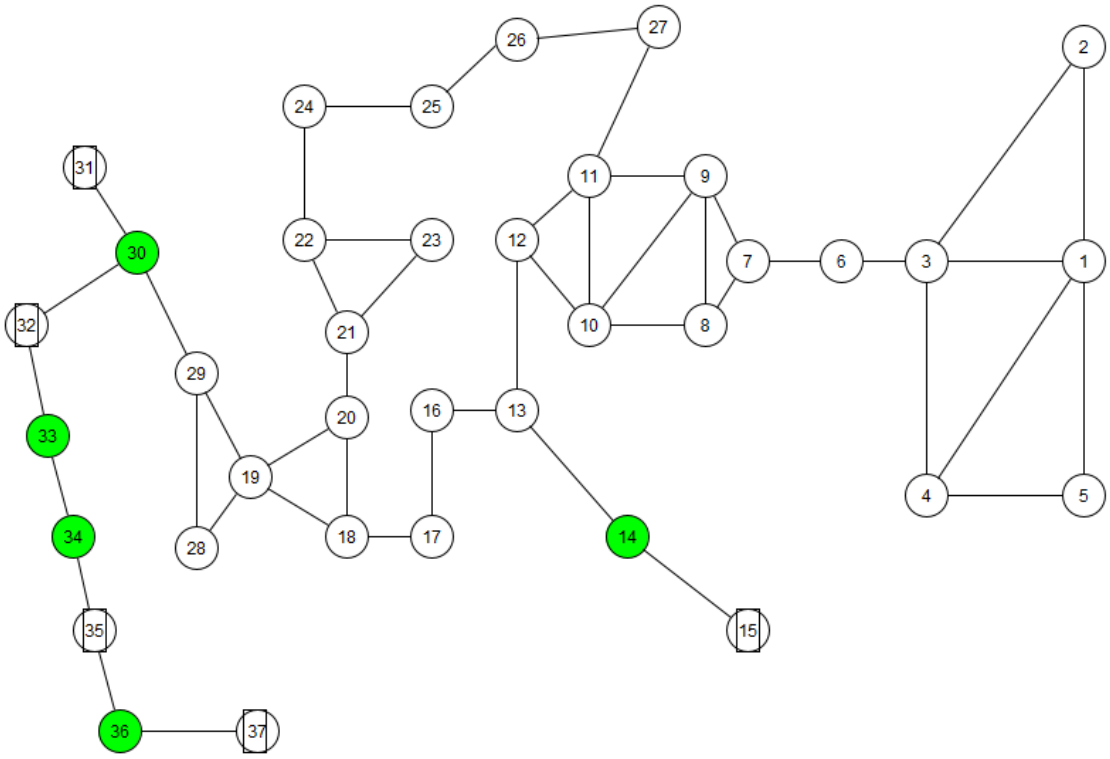

Figure 21: step2

Step3: We choose the nodes those have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,37,31,35,32,2,5,13,17,27,24,29}.
Critical = {14,36,30,34,33,1,3,4,11,12,26,25,22,16,18,19,28}.
Remain_Nodes ={6, 7, 8, 9, 10, 20, 21, 23}.

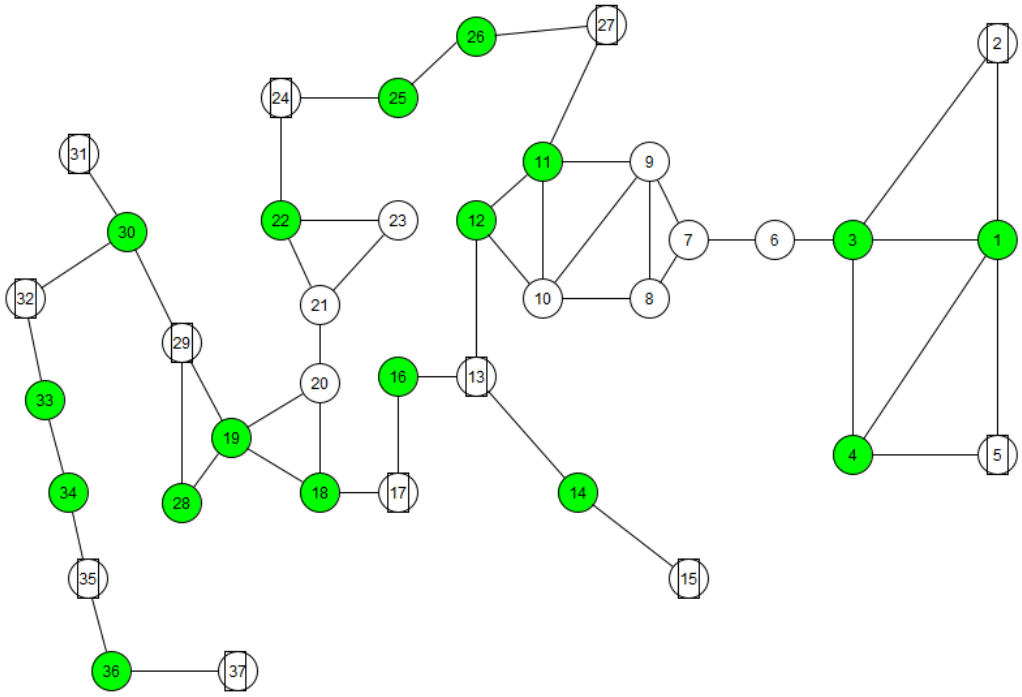

Figure 22: step3

Step4: We choose the nodes those have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,37,31,35,32,2,5,13,17,27,24,29,6,20,23}.
Critical = {14,36,30,34,33,1,3,4,11,12,26,25,22,16,18,19,28,7,21}.
Remain_Nodes ={9, 7, 8, 10}.

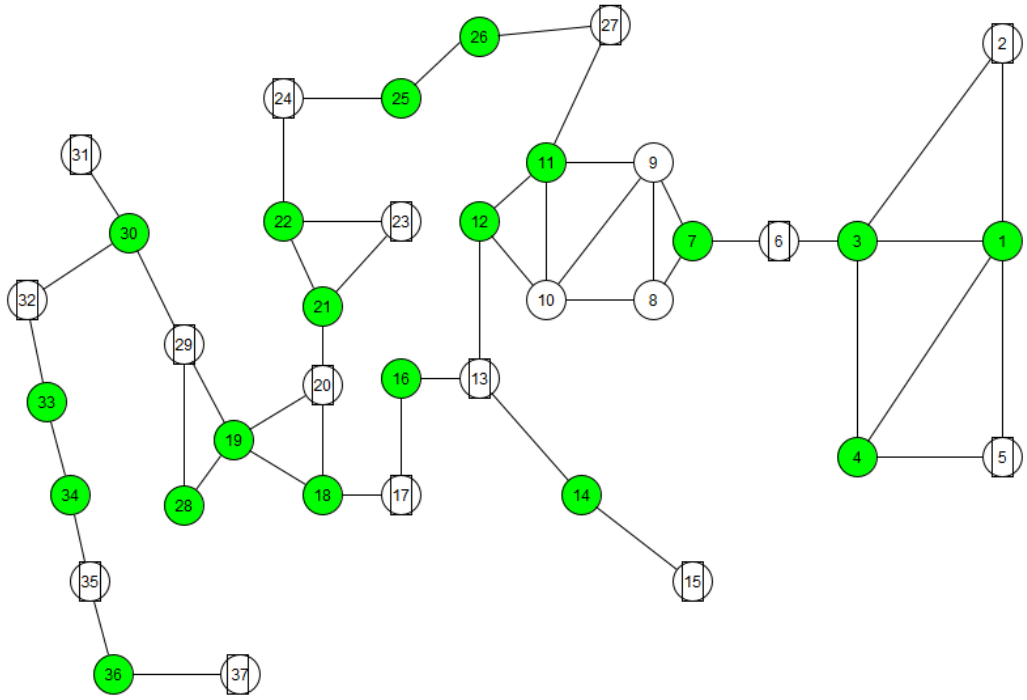

Figure 23: step4

Step5: We choose the nodes those have the minimum degree in the graph, we put it in the list of maximal independent set and we put its neighbors in the list of critical node (marking).

MIS= {15,37,31,35,32,2,5,13,17,27,24,29,6,20,23,9}.
Critical = {14,36,30,34,33,1,3,4,11,12,26,25,22,16,18,19,28,7,21,8,10}.
Remain_Nodes= Ø.

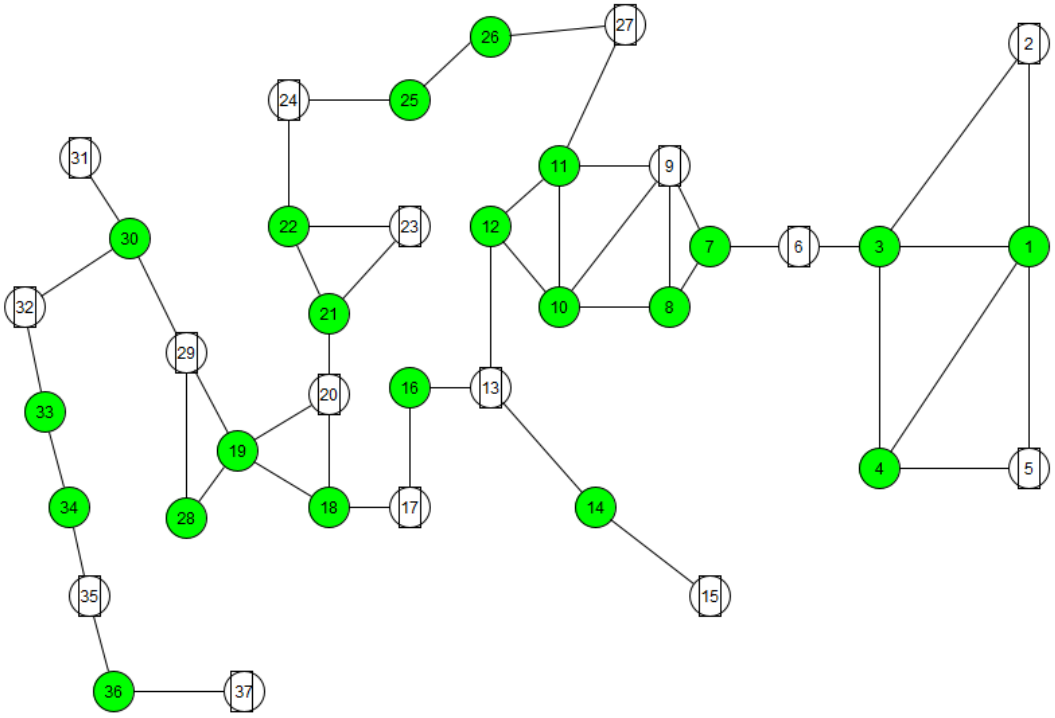

Figure 24: step5