PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

University Mohamed Seddik Ben Yahia of Jijel, Algeria

Faculty of Exact Sciences and Computer Science

Department of Computer Science

A Dissertation Submitted to the Department of Computer Science in Partial Fulfillment of the Requirements for the Degree of Master

Option: ARTIFICIAL INTELLIGENCE

Smart Traffic Light System For Smart Cities

Submitted by:
Bouzenoune Walid
Feraga Alaeddine

Supervised by:
Dr. Souici Ismahane

September 2021

Acknowledgment

Above all, We thank our Almighty God, who gave us strength, faith, health, will, and guidance to accomplish this modest work.

We wish particularly to acknowledge our Supervisor ***Dr. Ismahane Souici*** for having supervised, helped, guided, advised, and encouraged us throughout our work. We would like to praise the efforts and time that she devoted to us as a supervisor. Thanks to her instructions, the work has been adequately done.

We would also like to thank the jury members who have agreed to review and participate in the jury of our work.

We take this opportunity to express gratitude to our teachers of the computer science department at the University of Jijel, who have ensured our formation during our university cycle.

We would like to extend our sincere thanks to all those who have contributed in one way or another to the realization of this thesis.

Our deepest gratitude goes to our families, who have been able to approach us without letting go of the support during all these long years of study.

To all these contributors, We offer our thanks, respect, appreciation, and admiration.

Dedication

With deep gratitude and sincere words, I dedicate this humble work to my beloved and respectful parents, who have sacrificed their lives for my success, and who have given me their eternal and unconditional love, their support, encouragement, and their prayers, day and night, which enlightened my path and helped me to reach the highest expectations in life, hoping that one day, I will be able to give them back some of what they have done for me. May God grant them happiness, health, and long life.

To my dear brothers. To all my family that I am proud of.

I will never forget the help of my supportive supervisor, who believed in me and encouraged me until the end.

To all my dear friends.

Finally, to all those who have taught me throughout my school life.

Walid.

Dedication

With deep gratitude and sincere words, I dedicate this humble work to my beloved and respectful parents, who have sacrificed their lives for my success, and who have given me their eternal and unconditional love, their support, encouragement, and their prayers, day and night, which enlightened my path and helped me to reach the highest expectations in life, hoping that one day, I will be able to give them back some of what they have done for me. May God grant them happiness, health, and long life.

To my dear brother, and my dear sisters. To all my family that I'm proud of.

I will never forget the help of my supportive supervisor, who believed in me and encouraged me until the end.

Special credit would be devoted to a special person "The One", who supported me and gave me the strength to go on.

Alaeddine.

Abstract

Traditional traffic light systems were worked perfectly in the previous years. By the increase of population, vehicle number has grown significantly, which makes big problems over the world specially in crowded cities. The world development helps to create smart cities that allows to develop the traffic light systems to be smarter.

In this work we are trying to develop an intelligent system, based on Genetic Algorithm, that optimizes traffic light systems, and helps to reduce traffic congestion over four connected intersections. However, a problem of secure communication of the calculated genetic solution between the four intersections will be posed. The solution considered was the exploitation of blockchain technology to accomplish a secure verification and transfer of the combination illustrating the next road condition.

Keywords :*Smart cities, Congestion, Intelligent Traffic Light, Genetic Algorithm, Blockchain.*

Résumé

Les systèmes traditionnels de feux de circulation fonctionnaient parfaitement au cours des années précédentes. Avec l'augmentation de la population, le nombre de véhicules a augmenté de manière significative, ce qui pose de gros problèmes dans le monde, en particulier dans les villes surpeuplées. Le développement mondial aide à créer des villes intelligentes qui rendent les systèmes de feux de circulation plus intelligents.

Dans ce travail, nous essayons de développer un système intelligent, basé sur l'algorithme génétique qui optimise les systèmes de feux de circulation, et aide à réduire la congestion du trafic sur quatre intersections connectées.Toutefois, un problème de communication sécurisée de la solution génétique calculée entre les quatre intersections sera posé. La solution envisagée était l'exploitation de la technologie de blockchain pour accomplir une vérification et un transfert en toute sécurité de la combinaison illustrant le prochain état de la route.

Mots-clés*:villes intelligentes, congestion, Feu De Circulation Intelligent, Algorithme Genetique, Blockchain.*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI	Artificial Intelligence
TL	Traffic Light
ITL	Intelligent Traffic Light
ITLS	Intelligent Traffic Light Ssystems
GA	Genetic Algorithm
NN	Neural Network
GTL	Genetic Traffic Light
POS	Proof Of Stake
DPOS	Delegated Proof Of Stake
POW	Proof Of Work
SCOOT	Split Cycle Offset Optimisation Technique
SCATS	Sydney Coordinated Adaptive Traffic System
TRANSYT	Transportation Network Research Tool
TRRL	Transportation and Road Research Laboratory
PRODYN	PROgramation DYNamique
ITSA	Intelligent Traffic Signaling Agent
RSA	Road Segment Agent
EA	Evolutionary Algorithm
PKC	Public Key Cryptograghy

General Introduction

Most of traffic light systems around the world have a traditional mechanism that changes the light each time after giving a fixed time to the signals. And with the huge increase of population, that bring more use of private and public transportation, it makes a lot of use for roads every day, which is the thing that causes traffic congestion. The consequence of traffic congestion is represented in accident, loss of time, coast of money, etc. One of the solutions ensuring the resolution of such a problem is to build an intelligent traffic light system.
The world development makes the idea of building new intelligent traffic light systems (ITLS) possible, by creating smart cities. These systems allow the processing, by an artificial intelligent system, of data collected from detection systems. Many AI approaches such as fuzzy logic, intelligent agents, machine learning, artificial neural networks, genetic algorithms, etc, can be exploited to develop ITLS participating in solving the traffic congestion problem.
Genetic algorithms represent an important category of these methods. Their fields of application are constantly expanding given their multiple strengths (performance, speed, simplicity of their operators ...), where they are very successful especially in the field of artificial intelligence and operational research to solve optimization problems.
It is this artificial intelligence method that we used to develop an ITLS optimizing road condition across four connected intersections influencing each other. Thus, each intersection has an ITLS. Once the solution combination is calculated on one of the four ITLS, it is necessary to communicate that solution to the others to apply it. The transmission of such important information may be subject to attacks threatening its security (the identity of the origin and integrity of the information transmitted). Blockchain is one of the most promising technologies for satisfying such security.
Thus and apart from this introduction and the general conclusion which reproduces the work of all the chapters, the manuscript is divided into three chapters.
The first chapter exposes notions of smart cities, traditional and intelligent traffic light system and risks of smart cities. The same, a history about development of traffic light system, advantages and disadvantages of ITL and some classic management systems for traffic lights were presented. Finally, the use of artificial intelligence to design traffic light system was illustrated.
The second chapter presents an introduction to genetic algorithms. The vocabulary of this field, the different stages of the genetic process as well as the basic techniques used for the design of a simple, robust and efficient genetic algorithm were presented. Likewise, the key points of blockchain technologies as well as the cryptographic primitives involved have been summarized.
The third chapter describes the application of genetic algorithms to develop an ITLS starting from the adopted coding until arriving to calculate the solution optimizing the state of the roads on four intersections. Then, the outline of a secure transmission solution of this combination by exploiting blockchain technology was presented. The same chapter illustrates the parameterization of the developed genetic algorithm. In addition, results of application of the developed system, their interpretation and discussion were clarified.

CHAPTER 1

INTELLIGENT TRAFFIC LIGHT.

1.1 Introduction

The huge increase of population brings new issues to the cities; it needed to build more houses, parks, roads, etc. With this increase, people would absolutely need either private or public transportation, which makes an exponential growths of transportation that bring congestion, this one causes waste of time, stress and also pollution, etc.
This thing is the main cause for creating traffic light system, which basically started with a simple system used gas. By the development of cities and being smarter, traffic light systems have been developed to be intelligent.

In this chapter we will give definitions of smart cities, traditional and intelligent traffic light system, give some smart cities trends, mention some risks of smart cities, give a history about development of traffic light system, give some advantages and disadvantages of ITL, mention some Classic management systems for traffic lights and finally how to use artificial intelligence in traffic light system.

1.2 Smart city

1.2.1 Definitions

1. The Institute of Electrical and Electronics Engineers defines a smart city as a city that brings together technology, government and society to enable the following characteristics: smart economy, smart mobility, smart environment, smart people, smart living, and smart governance [4].

2. As a simplistic explanation, a smart city is a place where traditional networks and services are made more flexible, efficient, and sustainable with the use of information, digital, and telecommunication technologies to improve the city's operations for the benefit of its inhabitants [18].

1.2.2 Smart City Trends

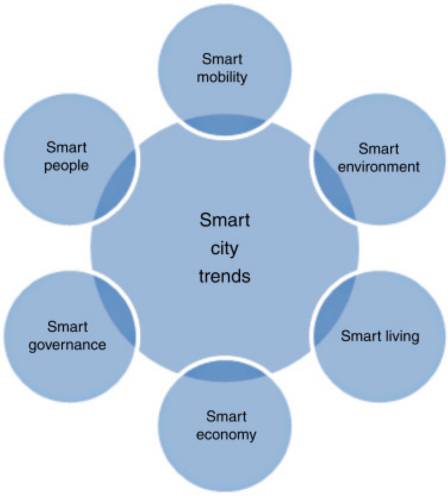


Figure 1.1: Smart city trends [50].

- Smart Mobility:
  - Local and national accessibility.
  - Safe and sustainable transportation systems for public transport.
  - Access to ICT supported and integrated transport.
- Smart Governance:
  - Involvement of the public in decision-making.
  - Public and social service.
  - Government Transparency.
- Smart Environment:
  - Judicious land use planning.
  - Attractiveness of natural conditions.
  - Environmental protection and pollution control.
- Smart Economy:
  - Entrepreneurship and productivity spirit.
  - An overall culture of innovation based on e-business & e-commerce.
  - Promotion of local products.
- Smart living:
  - Quality Health conditions.
  - Individual safety.

- Cultural and educational facilities.
- Accessibility to good quality housing.

- Smart People:
  - A culture of life-long learning.
  - Social and ethnic diversity.
  - Flexibility, Creativity & Community participation.
  - Citizen awareness.

1.3 Risks Of Smart Cities

- Privacy risks to citizens: Although data sharing between agencies may improve bureaucratic outcomes, data breaches at the federal level cast doubt on local governments' ability to keep data safe, and smart city tools also make possible the sale of personal data to private firms. Meanwhile, increasing governments' capacity for surveillance raises concerns about civil rights and privacy [17].
- Skewed incentives: Smart city tools also raise concerns about unhealthy incentives. For example, there is evidence that the statistics-driven policing tool CompStat motivates officers to patrol minority neighbourhoods where they focus on minor, easily prosecuted crimes. These strategies divert resources from more important but difficult cases and erode trust between police and the communities they serve [17].
- Cover for ineffective reforms: Politicians who lack the incentives to pursue meaningful change will often deploy technological "solutions" that fail to produce results. Examples in Washington. DC, and Philadelphia show that technological gimmicks can sometimes take the place of meaningful reforms [17].

1.4 Traffic light System

1.4.1 Definition Of Traffic Light

A traffic light is a device for regulating road traffic between road users and vehicles . The lights intended for vehicles are generally of the three-color type, to which directional arrows may be added. Those intended for pedestrians are two-tone and are often distinguished by the reproduction of a pedestrian silhouette.

1.4.2 History

The first traffic light invented was in London in the 1860's, but it was hardly recognizable or effective [13]. December 10, 1868 is the official birth date of the world's first traffic light. It was installed at the intersection of George and Bridge Streets near the Houses of Parliament in London. The system was composed of two mobile signs attached to pivoting

arms that were manipulated by a lever. The post was topped with a gas-lit semaphore to ensure visibility. But it was short-lived. Less than two months later, the traffic light exploded, killing the police officer who worked the signs.
In 1923, the first mechanical traffic light using electricity was installed in Paris at the intersection of Boulevard de Strasbourg and Grands Boulevards. Most of Europe's largest cities soon followed suit: Berlin in 1924, Milan in 1925, Rome in 1926, London in 1927, Prague in 1928, Barcelona in 1930. And the system was exported to Tokyo in 1931 [14].
The first Convention on the Unification of Road was signed in Geneva on March 30, 1931. Its goal was to increase road traffic safety and facilitate international movement by road through a uniform system of road signals. The majority of signs that we recognize today were defined through this treaty. Traffic lights with three colors (red, yellow, green) became the standard [14].

Today, new intelligent traffic lights manage traffic in Toronto (Canada). These assess road traffic and adjust the lighting time of the lights to allow smoother traffic and avoid traffic jams.
From its creation to the present day, its purpose has always been to control the circulation of vehicles. What has been changing (the only change) is how traffic lights manage such flow.

1.4.3 Areas Of Use

- Traffic management at intersections.
- The crossing of pedestrians, Operation by alternating one-way streets of a section where crossing is impossible or dangerous.
- Access control to certain expressways.
- The management of a checkpoint for people or vehicles needing to be stopped (toll).
- Protection of intermittent obstacles.

1.5 Traditional Traffic Light

1.5.1 Definition

The traditional traffic light systems basically operate with their static time base, that is, manually configuring the time at which each lamp will be lit. Its configuration can be either carried out by a professional operator or by a wired communication network, connecting each traffic light in the city to a single location, the so-called central management traffic light system [3].
The traditional traffic light is a kind of luminous devices and materials placed in crossroads and crosswalks for both people and vehicles to control traffic in a safe way, using colored LEDs according to a global system, which are:

- **Red light:** when you are driving and seeing red you have to completely stop and wait until it turns green, then you can proceed.

- **Yellow light:** there are two types of yellow light, solid and flashing. The solid one is simply a warning that means the red light is soon to follow, you got two choices, slow down and get ready to stop your vehicle, however if you were driving in a high speed, you can speed up and get through the intersection before the light changes. However, when the yellow light is flashing you are always allowed to drive right on through, you only have to reduce your speed and be cautious when crossing the intersection.
- **Green light:** it simply means « GO! », the road is yours, but before that, takes a second look both ways to make sure that it's clear.

As we know, the main purpose of creating traffic light is to control traffic, reduce congestion...etc.

1.5.2 Advantages Of Traditional Traffic Light

- Provide for orderly movement of traffic.
- Increase traffic-handling capacity of an intersection.
- Reduce frequency and severity of certain types of crashes, especially right-angle collisions.
- Provide continuous movement of traffic at a definite speed along a given route.
- Interrupt heavy traffic at intervals to permit other vehicles or pedestrians to cross.

1.5.3 Disadvantages Of Traditional Traffic Light

- Increased traffic congestion, air pollution and gasoline consumption.
- Disobedience of signals.
- Increased use of less-adequate streets to avoid traffic signals.
- Increased frequency of crashes, especially rear-end collisions.
- Wasting time to wait the green light even if the road is clear.

In crowded cities, it is difficult to implement a normal traffic control system, because of the variation of flow of roads vehicles during different period of time. Many parameters must be considered to develop a certain traffic control system. These parameters are concentrated on flow of vehicles, the emergency vehicles, the rush hours, the accidents, the important persons and the closing of any incoming road [10].

1.5.4 Classic Management System For Traffic Light

Usually, urban traffic control system takes several signal nodes as one sub-area based on the experiential datum such as changes of the traffic flow in the adjacent intersection, traffic dispersion and the number of junction links [5].

1.5.4.1 TRANSYT

The first commercial model is TRANSYT (Transportation Network Research Tool), which is a program for optimizing traffic signal control at a fixed time. For a network composed of multiple road segments and intersections, the program determines the traffic light plan (the best distribution of the green duration between different branches of all intersections and the best offset between them during the characteristic time period that flows into the network). These same connection points lead to optimal network operation. All traffic lights in the network run on the same cycle.

The first version of TRANSYT dates back to 1967 and was developed by the UK's TRRL (Transportation and Road Research Laboratory). Since then, the traffic signal plan calculated by TRANSYT is the basis of many traffic control systems implemented in many British cities and around the world. TRANSYT has also become a reference system for evaluating the effectiveness of real-time traffic control systems. It is constantly being improved, and its latest version (new release of the 10th edition) dates back to 1996 [17].

1.5.4.2 SCOOT

SCOOT is a centralized adaptive traffic signal control system developed in the UK in the early 1980s by the Transport Research Laboratory. The system has been modified and enhanced several times since its inception. The recent version of SCOOT is “Managing Congestion, Communications and Control” or MC3.

SCOOT, often called an online version of TRANSYT signal optimization tool, continuously measures the traffic demand on all roads in a coordinated network and optimizes signal timings for detected traffic. SCOOT uses an on-line model of traffic behavior, based on data from on-street detection, to estimate stops and delays within the network for the next cycle of traffic lights. The basic philosophy of SCOOT is to modify existing traffic signal settings in such a way that disturbance to traffic is minimal. Thus, the optimization routines are limited to evaluating only the impacts of small changes in signal settings on the estimated performance measures (delays and stops). Signal timings such as splits, offsets, and cycle lengths are optimized separately [7].

1.5.4.3 SCATS

SCATS was developed in Australia in the early 1970s and has been used successfully in Australia for the past 40 years. The current system uses loop detection near the stop bars in addition to video cameras to operate in real-time conditions. The system optimizes cycle lengths, phase splits, and offsets on a cycle-by-cycle basis. The degree of saturation is used to adjust the cycle length. Phase splits are timed by giving each approach an equal degree of saturation, or higher priority can be given to the main road. SCATS selects offsets on the basis of free-flow travel time and degree of saturation, which provides minimum stops for the vehicles on the main road-way. The popularity of SCATS has grown over time, and its use has expanded to other countries, including the United States.

SCATS has been installed in cities across the United States with mixed results. Various before-and-after studies have tested the improvements of adaptive traffic signal control compared with existing pretimed or time-of-day plans. Many claims are made about performance improvements; however, the results vary on a case-by-case basis. Some differences in performance improvements may be partly related to how the evaluation was conducted, in addition to other potential site-specific reasons [8].

1.5.4.4 PRODYN

PRODYN (PROgrammation DYNamique) is a traffic adaptive and decentralized system developed by French CERT (Centre d'Etude et de Recherche de Toulouse) in 1980s. In this system, 2 to 3 "electromagnetic" circuits are placed on each road segment, and a simple flow model is used to estimate the assumed traffic conditions on each lane, so that the traffic on the lane can be predicted. The system performs optimization on "isolated" intersections, but some versions of the system allow communication between adjacent intersections to predict incoming traffic. The strategy used by the system includes analysing whether switching the state of the traffic light (i.e. changing the phase) is the best decision at each time step (5 seconds), that is, whether to minimize the waiting time of the vehicle in front of the intersection according to the traffic model used, the next 75 seconds [17].

1.6 Intelligent Traffic Light

1.6.1 Definition

The intelligent traffic light system is a combination of traditional traffic light system equipped by an array of sensors, camera video detection system and artificial intelligence to intelligently route vehicle and pedestrian traffic.

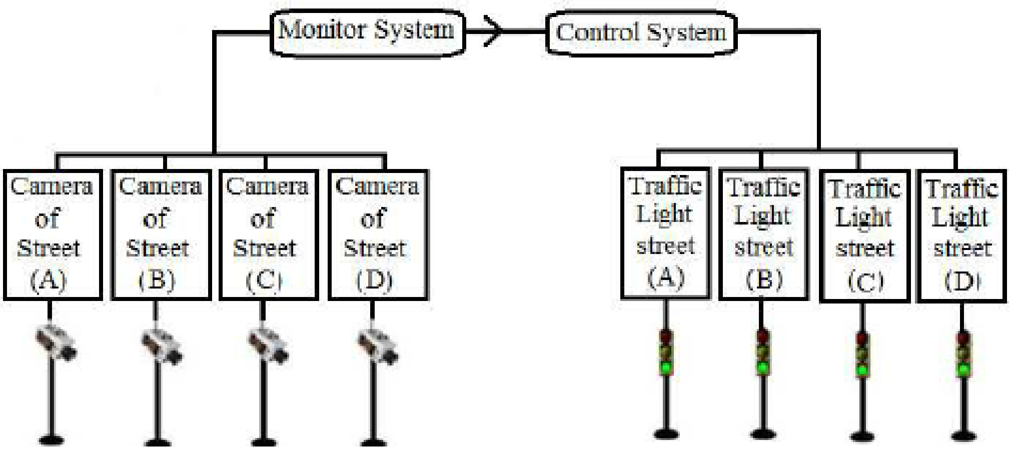


Figure 1.2: Intelligent Traffic Light system.

1.6.2 ITL Using Artificial Intelligence

Various computational intelligence-based approaches have been proposed for designing real-time traffic signal controllers, such as fuzzy sets [2], [3], genetic algorithm and reinforcement learning [4], and neural networks (NN) [5]–[7]. Most of these works are based on the distributed approach, where an agent is assigned to update the traffic signals of a single intersection based on the traffic flow in all the approaches of that intersection [16], we mentioned the following:

A Fuzzy Logic: The first known attempt to apply fuzzy logic in traffic control was made by Pappis, and Mamdani. They simulated an isolated signalized intersection composed of two one-way streets with two lanes in each direction without turning traffic. The fuzzy controller reduced average vehicle delay compared to an actuated controller [11]. Many efforts have been done relaying in Fuzzy Logic, because it approximates how human beings think, giving more advantages than the Boolean logic. Furthermore, it is easy to implement. Most of ITS systems that implement Fuzzy Logic are practical in a single intersection [1].

B Intelligent Agents: Traffic lights can be modeled as intelligent agents that communicate with each other to improve traffic conditions of the whole network [1].

Roosemond (1998) describes intelligent agent architecture for traffic light control. Intelligent traffic signaling agents (ITSAs) and Road Segment Agents (RSAs) try to perform their own tasks, and try to achieve local optimality. One or more Authority Agents can communicate with groups of ITSAs and RSAs for global performance. All agents act upon beliefs, desires, and capabilities [2].

E Genetic Algorithms: In 1960 the genetic algorithm was proposed by John Holland. The goal is to find the right solution for complex and large search problems. This is the advantage of a genetic algorithm that belongs to the larger class of evolutionary algorithms (EA). This algorithm often succeeds in obtaining the best ideal solution so it has become preferred to use. The genetic algorithm has several advantages such as explored solution space multiple directions at once [15].

1.7 Conclusion

In this chapter, an overview of the intelligent traffic light system is presented. Initially we have presented a definition of smart city, its trends and its risks. After we have defined the traditional traffic light system, how it works and its advantages and disadvantages. Next we mentioned some classic management systems for traffic lights. After that we have talked about the intelligent traffic light system and how it uses AI. Finally some intelligence-based approaches for building an ITL were described, such as Genetic algorithm that we are going to implement in our work. So in the next chapter, we will give a background about the GA and the blockchain technology for securing our system.

CHAPTER 2

BACKGROUND

2.1 Introduction

For a long time, information and communication technology has made numerous developments in order to promote, strengthen and ensure the exchange and sharing of information, data and funds in various ways. With the advent of the Internet, digital communications came into being, realizing various forms of data exchange through online transactions. The development of the Internet has brought important security issues and challenges, as well as corresponding countermeasures. These issues have an increasing impact on trust. Trust is the cornerstone of our society, because every human interaction uses trust as a medium, and the information society also needs trust to continue. It needs a kind of digital trust that should be realized by information technology.

In this chapter, we'll try to give an overview of genetic algorithms, its important definitions and terms, its basic process and the tasks that must be realized. We'll also try to introduce Blockchain, a new and creative technology. We will demonstrate how this technology may be used to securely communicate and control information among parties who don't necessarily trust each other, as well as how it can improve the way we deal with transactions. We'll also explain its concept and architecture.

2.2 Genetic Algorithm

2.2.1 Presentation

Genetic algorithms (GA) are an optimization algorithm, which means that they are used to find the best solution for a given computational problem that maximizes or minimizes a specific function. Genetic algorithms represent a branch of the research field called evolutionary computing because they mimic the biological process of reproduction and natural selection to solve the "fittest" solution. Just like in evolution, many processes of genetic algorithms are random, but this optimization technique allows setting the level of randomization and control. These algorithms are more powerful and effective than random search and exhaustive search algorithms, but do not require additional information about a given problem. This feature allows them to find solutions to problems that other optimization methods cannot handle due to lack of continuity, derivative, linearity, or other characteristics[20].

GAs are successfully applied to several optimization problems in several disciplines that are difficult to solve by classical mathematical programming [31, 32].

In the following sections, some important terminology and concepts of GA are presented.

2.2.1.1 Genes And Chromosomes

Gene is the basic part of genetic algorithm. A string of genes is called a chromosome. Chromosomes can be encoded as binary strings, real strings, etc.

2.2.1.2 Populations And Generations

A population is a set of chromosomes. GA starts with a set of randomly created individuals (chromosomes). This set is called the initial population. The iterations of GA are called generations. Each iteration involves selecting individuals with closely related characteristics and recombining them until a new generation is created to replace the old.

2.2.1.3 Parents And Children

The selection of chromosomes from one generation to another involves the selection of individuals with probabilistic methods [26]. Those with high fitness values have a high probability of being selected for crossover and producing new chromosomes called offspring or offspring. The a priori fixed probability of occurrence of crossover is called the crossover rate. It involves randomly selecting the intersection of parental chromosomes, where the genetic information of the parents should be mixed.

2.2.1.4 Mutation

Mutation is the process of introducing many new points into the search space. It ensures that positive choices do not lead to sub-optimal solutions. In other words, it can prevent premature convergence to the local optimum. It is achieved by randomly changing some chromosomal characteristics, and is performed with a very low probability value.

2.2.1.5 Fitness

The objective function that defines the goal of optimization is called the fitness function. It means everyone's "good" or "bad".

2.2.1.6 Elitism

In order to improve the performance of GA, the best individuals must always participate in reproduction. However, if these individuals are destroyed by crossover or mutation operators, they may be lost. Therefore, the first one is best Chromosomes or the few best chromosomes are copied into the new population.

2.2.2 A Basic Genetic Algorithm

The basic components common to almost all genetic algorithms are :

- Fitness function for optimization.
- Population of chromosomes.
- Selection of which chromosomes will be reproduced.
- Crossover to produce next generation of chromosomes.
- Random mutation of chromosomes in new generation.

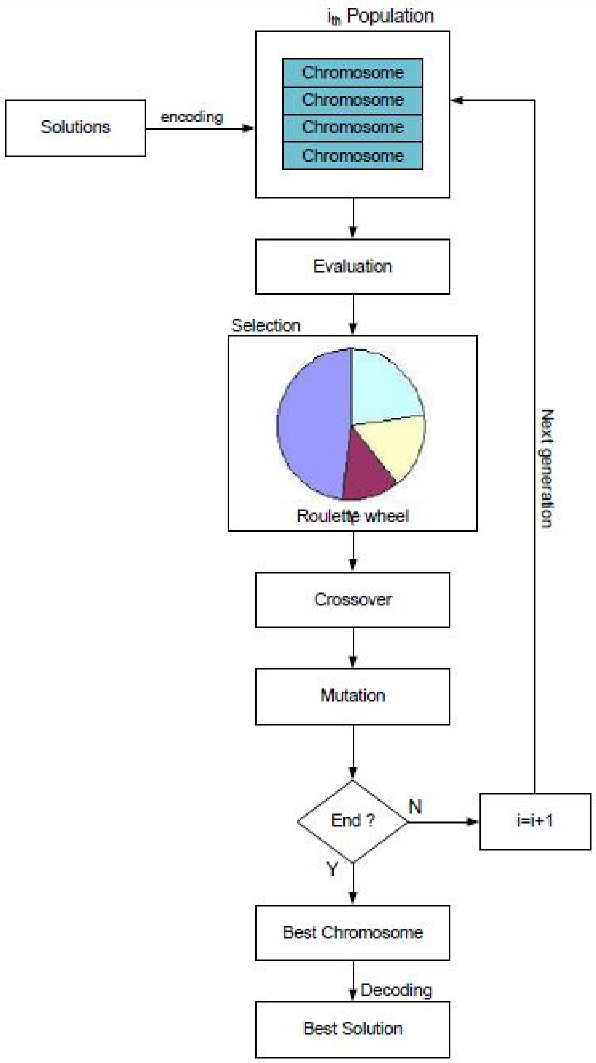


Figure 2.1: Flow chart of AG [21].

All of the following steps present a reading of the contents of the figure 3.1:

Step1. Determine the way of encoding, means the way of representing the individual genes of the chromosome initial, and that depends on the problem we are going to solve.

Step2. Generate the initial chromosome, and then the initial population from the initial chromosome in a random way, generally.

Step3. Process steps 4-7 until the number of generations is met .

Step4. Evaluation of fitness value of chromosomes by function calculating objective.

Step5. Chromosomes selection .

Step6. Crossover.

Step7. Mutation.

Step8. Solution (Best Chromosomes).

Note: We have to determine the number of chromosomes in the population. And mutation rate and crossover rate value.

The basic principles of a GA were shown in Figure 3.1, but as usual, the details are all important. The various stages involved in implementing a GA will now be described.

2.2.3 How Does GA Work?

The standard GA first needs to encode all the parameters of the optimization problem as a finite length chain. The principle of GA is very simple. It simulates the evolution of a group of individuals until a stopping criterion is met.

GA starts with the generation of the initial population and the evaluation of the adaptive functions of all individuals that make up the initial population. Then, individuals are randomly selected for reproduction according to the most suitable survival principle. Then, generate individual "children" (or offspring) by applying the following two genetic operators: Crossover and mutation. These children are placed in a new P(t) population and will replace all or part of the previous generation population. The new individual population will follow one generation (t) to one generation (t+1) until the judgment criterion is met.

2.2.3.1 Encoding

Each parameter of a solution is assimilated to a gene, all the values it can take are the alleles of that gene, we must find a way to code each allele differently in a unique way. A chromosome is a sequence of genes, for example, we can choose to group similar parameters in the same chromosome and each gene will be identifiable by its position. Each individual is represented by a set of chromosomes, and a population is a set of individuals.[22]

There are two main types of encoding that can be used, and you can switch from one to the other, plus or less, easily: the binary encoding, the real number encoding.

There are other different kinds of encoding, like hexadecimal encoding, octal encoding, tree encoding, etc [29].

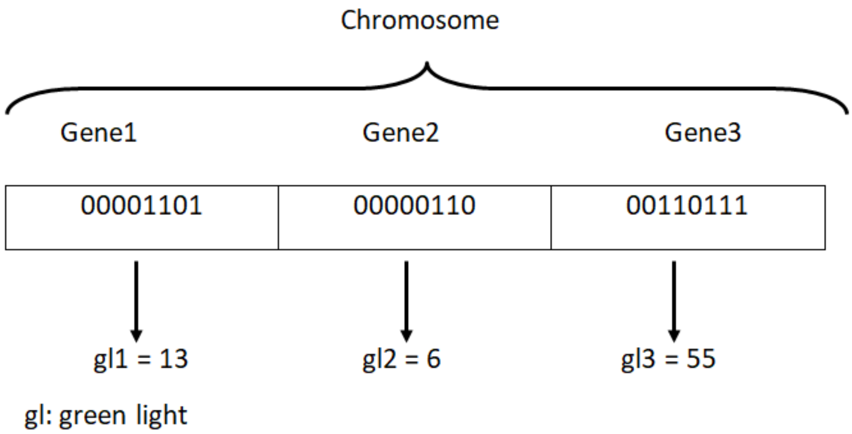


Figure 2.2: Coding real variables of a chromosome using binary encoding.

2.2.3.2 Selection

Selection involves selecting the best individuals for crossover. It aims to take advantage of the good characteristics of these individuals by considering their fitness value, which is a measure of "goodness". In theory, there are many selection strategies; however, the most commonly used options are described below [29].

- **Population decimation:** The simplest deterministic strategy is population decimation. Here, individuals are ranked from large to small according to their fitness value. An arbitrary choose the minimum fitness as the cut-off point, and remove any individuals whose fitness is lower than the minimum from the population. The remaining individuals are then used to generate a new generation through random pairing. Repeat the pairing and application of the GA operators until the new generation is filled [46].

  The advantage of population decimation selection lies in its simplicity. However, its disadvantage is that once an individual is removed from the population, any unique characteristics of the population owned by the individual will be lost [46].

Individual	Chromosome	Fitness
I1	00010111	24
I2	00101100	32
I3	10010110	15
I4	01011011	20
I5	00101100	5

Rank	Individual	Chromosome	Fitness
1	I2	00101100	32
2	I1	00010111	24
3	I4	01011011	20
4	I3	10010110	15
5	I5	00101100	5

Rank	Individual	Chromosome	Fitness	
1	I2	00101100	32	Keep
2	I1	00010111	24	Keep
3	I4	01011011	20	Keep
4	I3	10010110	15	Discard
5	I5	00101100	5	Discard

Figure 2.3: An illustrative example of the population decimation.

- **Proportionate Selection:** This method is also known as the roulette wheel. Its idea is to select individuals based on the selection probability given by the equation below [25, 26]:

$$P_i = \frac{f(parent_i)}{\sum_j f(parent_j)}$$

Where :

- P_i : is the probability of an individual parent being selected.
- $f(parent_i)$: is the fitness value of parent i.

By using the roulette wheel method, the individual that have the better weight is the one who has more chances to be selected.

The roulette wheel method principle resembles that of casino roulette, in that it amounts to imagining a kind of roulette casino on which are placed all chromosomes of the population according to their fitness values, then the ball is thrown and stops on a chromosome, and it's going to be the selected chromosome.

Roulette wheel selection is easier to implement but is noisy. The rate of evolution depends on the variance of fitness in the population[29].

The figure below represents the proportionate-selection process as a roulette wheel, where individuals are assigned a space on the wheel that is proportional to their relative fitness. The wheel is "spun," and the individual pointed to at the end of the spin is the individual selected.

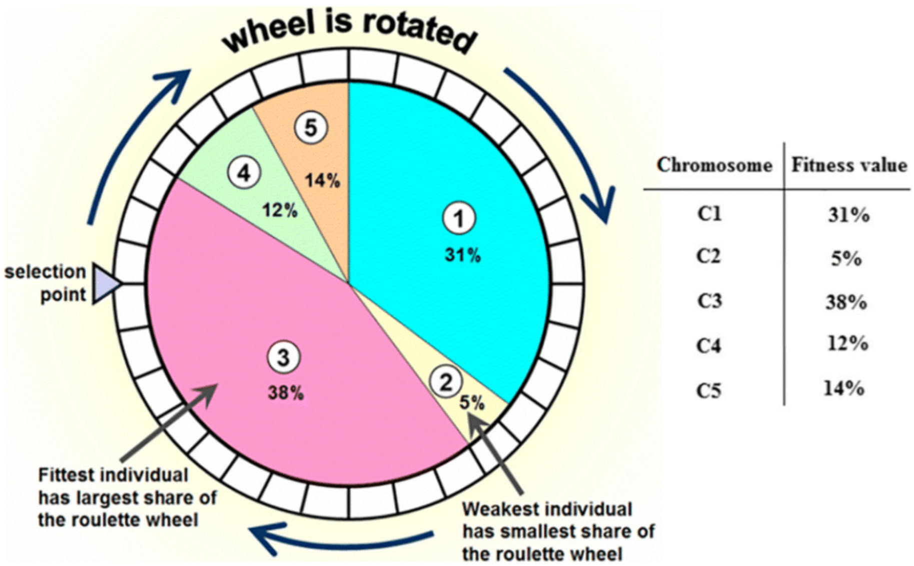


Figure 2.4: Proportionate selection represented as a roulette wheel [45].

- **Tournament Selection:** The ideal selection strategy should be able to adjust its selection pressure and population diversity to fine-tune GA search performance. Unlike roulette selection, the tournament selection strategy provides selection pressure by holding tournament matches between N_u individuals.

The best individual in the competition is the one with the highest physical fitness, that is, the winner of N_u. The tournament game and the winner are then inserted into the mating pool. The game is repeated until the mating pool that produces new offspring is filled. The mating pool composed of tournament winners has a higher average population fitness. Fitness difference provides selection pressure, it drives the genetic algorithm to improve the fitness of subsequent genes. This method is more effective and leads to the best solution [29].
There are other selection's methods such as [29]:

- Rank selection.
- Random selection.
- Boltzmann Selection.
- Stochastic Universal Sampling, etc.

2.2.3.3 Crossover

The crossover involves the birth of two new children from two parents. These schemes differ from binary to real number coding.

- **Binary GA Crossover:** For genetic algorithms with binary coding, there are multiple ways to perform crossover. The two selected parents simply exchange parts of their chromosomal structure based on random set points called crossover sites. The programmer chooses the number of exchange points [24]. We distinguish between two crossover schemes:

  - **Single point crossover:** Here, two parents exchange their parts at one single point. The example in Figure 2.5 illustrates how two 8-bit parents can use single-point crossover to produce two new children.

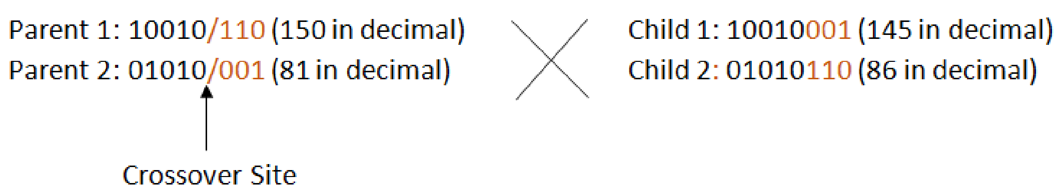


Figure 2.5: Single point crossover.

  - **Multiple point crossover:** In this scheme, the two parents interchange their parts at many locations. Figure 2.6.a and Figure 2.6.b give examples of a two-point crossover and a three-point crossover.

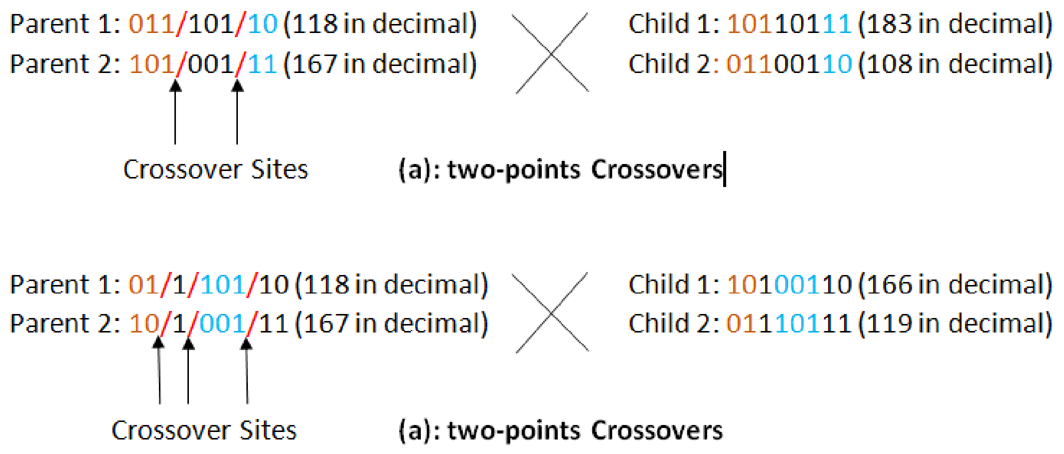


Figure 2.6: An illustrative example of multiple point crossovers.

- **Real coded GA Crossover:** In real code GA, crossover involves taking part of the features provided by the first parent and completing another part with the features of the other parent. These features are selected at random. The newly created child will be a combination of the attributes of its parents. The equations controlling this process are [25]:

$$P_{new1} = B * Parent_1 + (1 - B) * Parent_2$$

And

$$P_{new2} = (1 - B) * Parent_1 + B * Parent_2$$

Where: B is a random number$(0 \leq \text{B} \leq 1)$

2.2.3.4 Mutation

The role of this operator is to change randomly with a certain probability, In general, the probability of mutation is very small (typically between 1% and 7%) to avoid losing the good properties of the chromosomes [26].

A mutation consists, in its simple application, on the inversion of a bit (or several bits, but given the probability of mutation it is extremely rare) being in a very particular locus and also determined randomly.

The mutation operator therefore changes the characteristics of a solution in a completely random manner, which makes it possible to introduce and to maintain the diversity within our population of solutions. This operator plays the role of a "disruptive element" it introduces "noise" into the population [23].

several mutation operators have been proposed [40]:

- **Inversion mutation:** randomly select two positions within the chromosome, and then reverse the substring between these two positions.

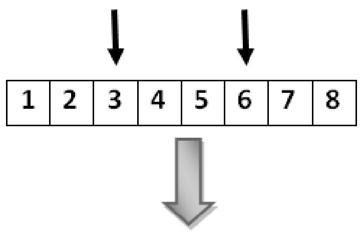


Figure 2.7: An illustrative example of inversion mutation.

- **Insertion mutation:** Randomly select a gene and insert it into a random position.

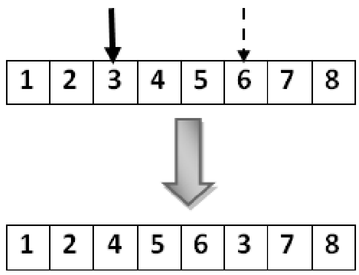


Figure 2.8: An illustrative example of insertion mutation.

- **Displacement mutation:** randomly select a substring of the gene and insert it in a random position. Therefore, insertion can be viewed as a special case of displacement.

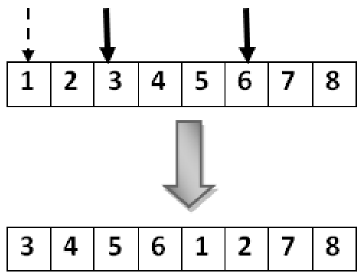


Figure 2.9: An illustrative example of displacement mutation.

- **Reciprocal exchange mutation (swap mutation):** randomly select two positions, and then swap the genes on the positions.

2.2.3.5 Stopping Criteria

Generally, the generation and replacement cycle is repeated until stopping criteria is met. This criteria may include a fixed number of iterations (generations), a maximum calculation time or/and a satisfactory solution.

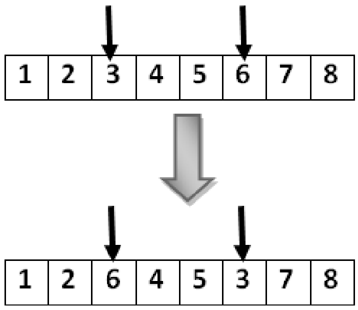


Figure 2.10: An illustrative example of reciprocal exchange mutations (swap mutation).

The genetic algorithm then returns the best solution(s) has been identified from generation to generation [27].

2.2.4 Some Application Of Genetic Algorithms

Why try to use GA instead of methods that are more traditional? The answer to this question is simple. GA has proven that it can solve many large and complex problems that other methods have encountered difficulties.

There are many practical problems and areas where genetic algorithms are successfully applied. They include: [26].

- Image processing.
- Laser technology.
- Medicine.
- Robotics.
- The architectural aspects of building design.
- Facial recognition.
- Training and designing artificial intelligence systems such as artificial neural networks.
- Aeronautics, etc.

2.2.5 Advantages And Disadvantages Of GA

2.2.5.1 Advantages

- They are adaptable to several types of problems.
- GA has a high robustness, i.e., a high ability to find the global optimizations problems.
- Easy to implement.
- Finally, GA is easily parallelised [28].

2.2.5.2 Disadvantages

- Compared to more traditional optimization methods, GA is still not very effective in terms of cost (or convergence speed).
- Sometimes GA will quickly converge to specific individuals with very high fitness values in the population.
- It is sometimes problematic that solutions encoded in bit string conform to domain constraints. You need to choose the encoding and even modify the operator [28].

2.3 Blockchain

2.3.1 Blockchain Concept

In 2008, an anonymous individual or group named Satoshi Nakamoto proposed and deployed blockchain technology for the first time. Satoshi Nakamoto developed a decentralized peer-to-peer electronic cash system that uses a new technology (later called "blockchain") to create Bitcoin, which is the famous and controversial cryptocurrency. The proposed system will allow online payments to be sent directly from one party to the other without going through a financial institution mechanism. Bitcoin is able to create a decentralized environment in which cryptographically verified transactions and data are not controlled by any central agency or intermediary. [43]

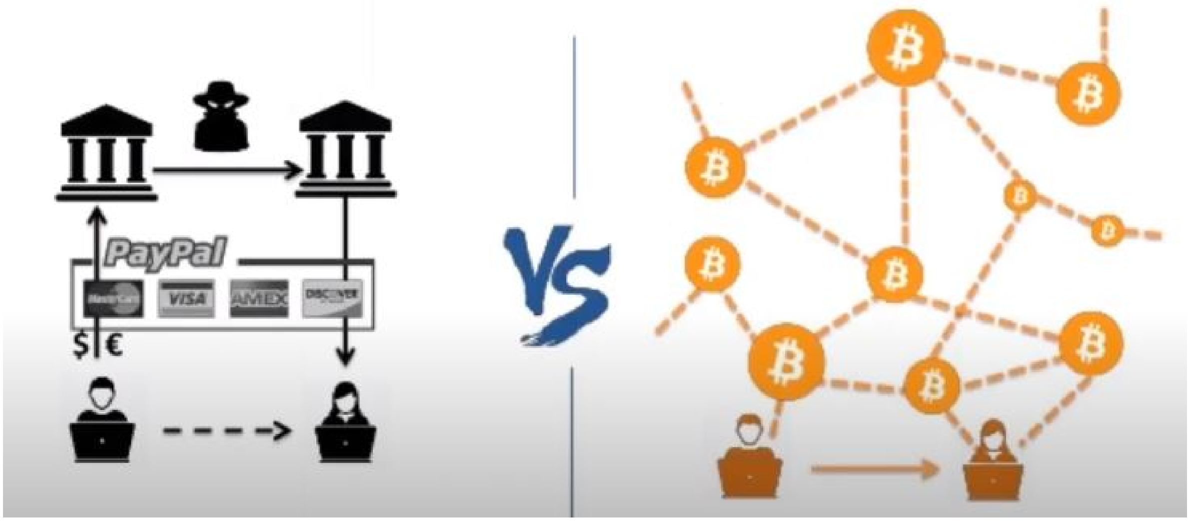


Figure 2.11: Comparaison between classic system and system based on blockchain.

2.3.1.1 Definition

A blockchain is a chain of blocks (data records) that are distributed across a decentralized network of computers, meaning that there is no central authority not a single point of failure, while each block contains some data which is depends on the type of the blockchain, the hash of the block and the hash of the previous block. Each node (computer) in the network stores and maintains an entire copy of the ledger (blocks), and any update requires the approval of the majority of the nodes in the network, which make it hard or let's say impossible to manipulate. [44]

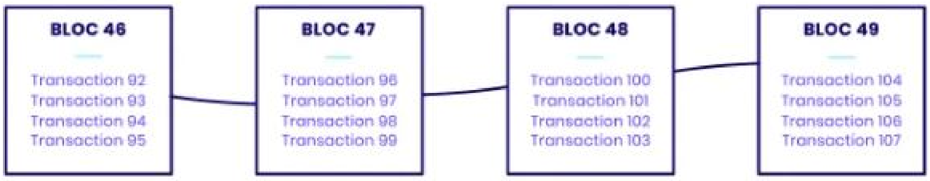


Figure 2.12: structure of blockchain.

2.3.1.2 Underlying technologies

Cryptographic Concepts: Cryptography is the science of secret communication, it is the technique which is used for doing secure communication between two parties in the public environment where unauthorized users and malicious attackers are present [47].

Blockchain uses mainly the following cryptographic functions: cryptographic hash functions, public key encryption and digital signature.

- **Hash functions:** The hash function is a recent term of modern computer science. Hash functions are used to generate some sort of fingerprint of data. The generated fingerprint is used to determine if the data has changed or not. This properly is useful to achieve data Integrity goal. If the data is altered, the fingerprint will change too. It is a type of one-way function that takes an arbitrary input of any number of bits and outputs a fixed n number of bits. This output of one-way function must possess properties like, it must be computationally infeasible to get back the input, and it must be computationally infeasible to find a pair of messages which has same hash value [48].

- **Public key cryptography (PKC):** Public key cryptography is one of the biggest foundations of network security. It uses two different keys to encrypt and decrypt the data. These two different keys are mathematically related. One is the public key and the other is the private key, they come as a pairs. The public key encryption is also known asymmetric key encryption because two different keys are use. In public key encryption, the Public key is public to anyone, while the private key belongs only to the person who creates these two keys [49].

- **digital signature:** Public key cryptography is also used together with hash functions to sign data. The concept is called digital signature and it allows authenticate data and its origin.

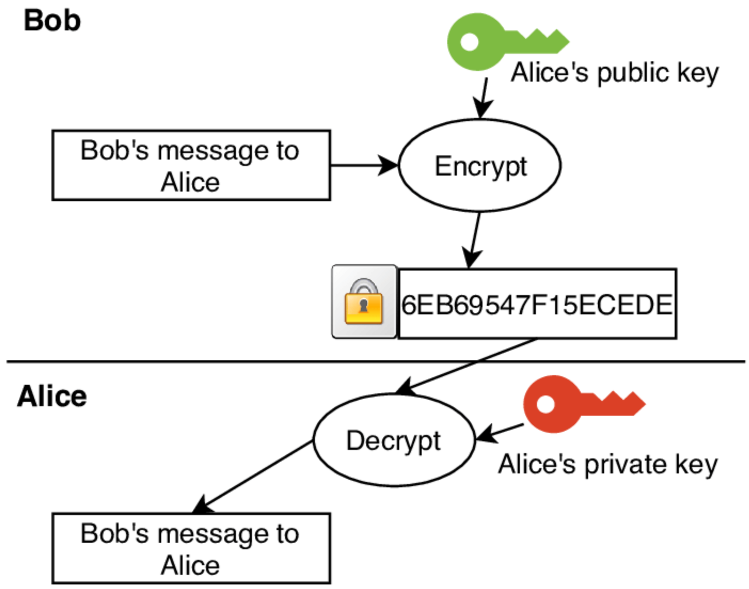


Figure 2.13: Public key cryptography.

2.3.1.3 Characteristics Of Blockchain

- **Decentralized:** One of the core aspects of blockchain is that it is a decentralized ledger, which means that data is maintained and held by all nodes in the network. There is no central authority to hold or update the ledger. In addition, every peer in the system has the right to add new transactions. Every transaction that passes the consensus phase will be recorded on the ledger[35].

  The difference between centralized, decentralized and distributed systems is schematically shown in Figure 3.14

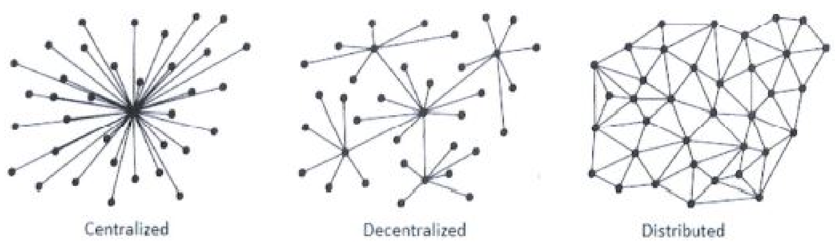


Figure 2.14: centralized, decentralized, distributed systems.

- **Immutable:** Once a transaction is added to the blockchain, it cannot be deleted or updated. This immutability is one of the main aspects that contribute to the trustworthiness of the blockchain system. The establishment of a consensus mechanism makes the system unable to cheat and makes it very reliable. The distributed ledger can be regarded as an irreversible pennant record [35].

- **Consensus Driven:** According to a well-known consensus process at the core of blockchain, no block may be added to the ledger without the approval of specific nodes in the network. [As a result, nodes may not trust one other, but they can trust the system as a whole]. Depending on the type or use case of blockchain, there are a lot of different consensus algorithms.

- **Transparent:** Because the ledger is shared across several network peers, any network user can see all transactions from the creation of the blockchain to the last recorded block.
- **Secure:** The blockchain is a cryptographically secure digital signature that verifies that the data in a block has not been tampered with.
- **Robust:** The way the blockchain works makes it an extremely robust technology. The benefit of a distributed network is that no user has access to more systems than another. If a peer is attacked, it will have no effect on the rest of the network. It makes no difference if one or more nodes depart the blockchain system.

2.3.2 Structure Of Blockchain

The blockchain technology consists of several components:

2.3.2.1 Block

Blocks are the basic data structure (file) in the blockchain, and they are linked together to form a blockchain. Each block can be considered as a page in the ledger. A block is a record of valid transactions that have not been recorded in the linked block. This process depends on the application; consider the Bitcoin example, where a block takes about 10 minutes to create and add to the blockchain, the Ethereum block only takes 14 to 15 seconds; some other blockchains create a new block every five seconds, and so on. A single block is composed of several components; almost these can be divided into a block header containing metadata and a block body [35].

2.3.2.2 Transaction

Transaction is the basic unit of blockchain. It represents the transfer of value from one account (address) to another. This transfer is spread across the network, collected by minors and included in the block. It is first sent to all connected nodes to increase the chance of being added to the block. To face the double-spending problem, we can only transfer unspent transactions.

In order to prevent each node from having to check the complete history of the blockchain to obtain some unspent transactions. By design, the transaction is either completely spent or unspent, which means that it is impossible to spend only part of the transaction. The remaining amount can be transferred back to your "wallet" to create a new unspent transaction.

A piece is defined by Nakamoto as a chain of digital signatures. During the transfer, the owner of the piece signs a hachage of previous transactions as well as the receiver's public key and adds it to the end of this chain of digital signatures. The private key is used to sign the transaction, whereas the public key is used to double-check the transaction [36].

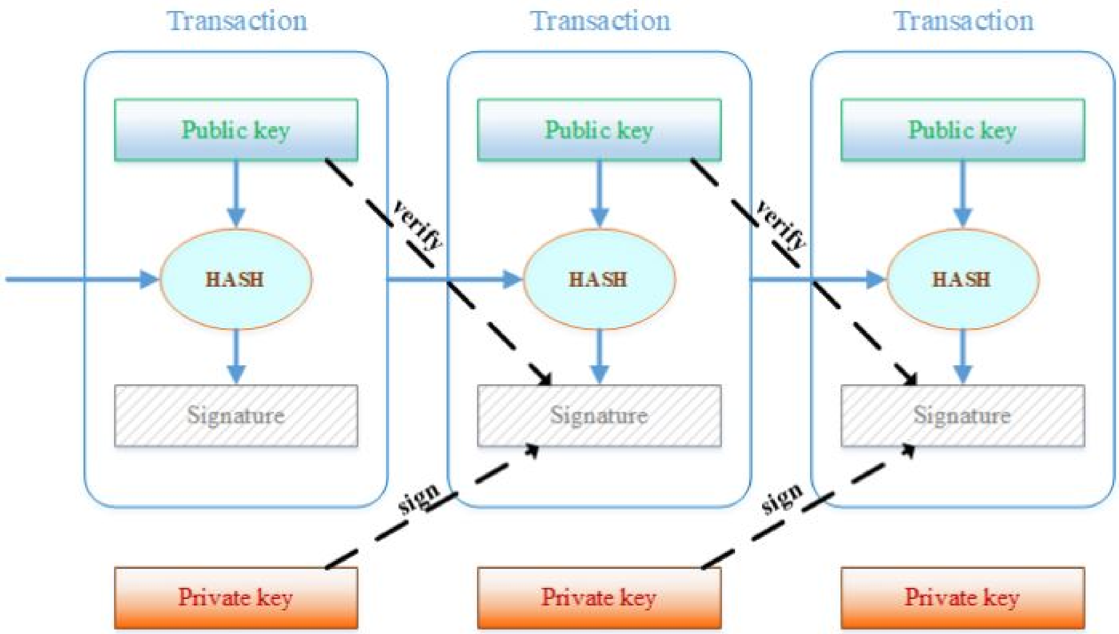


Figure 2.15: Structure of transaction [37].

2.3.2.3 Blockchain network

The blockchain network is composed of many nodes distributed around the world, and each node maintains a local copy of the blockchain, which contains a complete record of all transactions that have ever existed. It is a distributed peer-to-peer network in which each node can communicate with each other without central authorization. Nodes work hard to validate transactions in order to add blocks to the blockchain and earn rewards; they always assume that the main branch is the longest and continue to expand it. Before broadcasting a transaction to all other connected nodes, it should be confirmed first. Data passes from one node to the next in this way, eventually reaching the entire network.

2.3.2.4 Consensus

In the context of blockchain, consensus is critical. The idea is for everyone to agree on a single blockchain state. Because there is no central authority to determine whether new blocks are valid, each node must decide whether or not to accept a new received block [41].

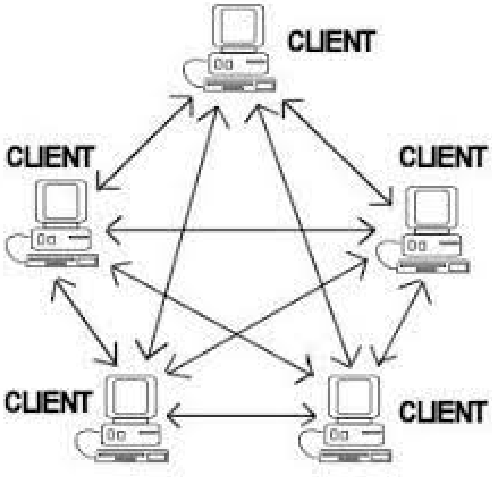


Figure 2.16: peer to peer network.

The goal of consensus process is also to make the network more robust against the various types of attacks. As it is mentioned above only one node will choose the block to validate [38].

It exists many different variants of consensus mechanisms (protocols) such as: proof of work (POW), proof of stake (POS), delegated proof of stake (DPOS), etc.

Besides those mechanisms there are dozens of other consensus mechanisms out there where each one of them satisfies a need and serves a purpose.

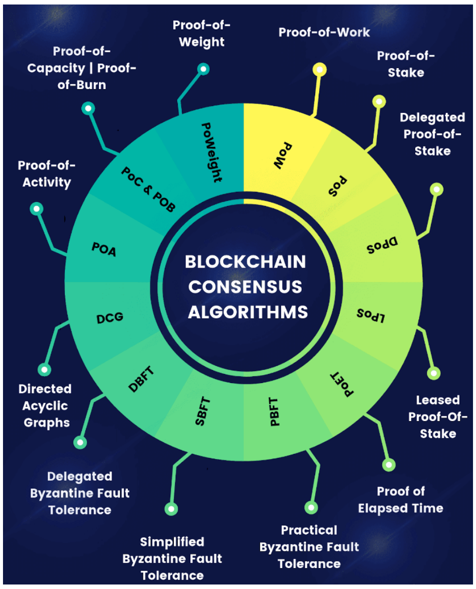


Figure 2.17: Overview on consensus mechanisms [42].

We'll detail below just the principle of the two most known and used consensus, POS and POW:

- **Proof Of Stake (POS):** Proof of stake is a consensus algorithm proposed in 2012 as an alternative algorithm for proof of work [44]. This mechanism is based on coin age rather than computing power. The coin age is simply defined as the number of currencies x the holding period. That means a miner who holds 3 coins for 10 days has a coin age of 30 days, and this coin age can only be obtained by holding coins [43]. Another concept of the Proof-of-Stake algorithm is to reach a consensus by requiring users to stake an amount of their stakes, so that they have the opportunity to be selected to verify the transaction block and then receive rewards.

- **Proof Of Work (POW):** The proof-of-work mechanism is considered to be the most famous consensus in the blockchain because it was used with the first cryptocurrency that never existed. Proof of work is a requirement that defines expensive calculations (also known as exploration) that must be performed to create a new set of unreliable transactions (blocks) on the distributed blockchain registry. The extraction process checks the legitimacy of the transaction, or avoids the so-called double spend, and creates a new digital currency by rewarding miners for performing the previous task [41]. Whenever a transaction is defined using the POW algorithm, the following events occur behind the scenes [40]:

  - Transactions are grouped in a block.
  - The minor verifies whether the transaction in each block is legal.
  - For this, miners must solve a mathematical problem called the proof-of-work problem, which can only be solved randomly (brute force).
  - Rewards are given to the first miner who solves the problem of each block.
  - Verified transactions are stored in the public blockchain.

  The major disadvantage of POW is the energy consumption, whose miners aim powerful computers for more computing power.

2.4 Conclusion

In this chapter, an overview of genetic algorithms and their appearance is provided. Initially, we gave important definitions and terms. Then, we introduced the basic process of genetic algorithm and the tasks that must be realized. Next, the genetic algorithm operator is introduced in detail, the stop criteria is given and we talked about some applications of GA and its advantages and disadvantages.
We also introduced the blockchain and its concept, which is a new revolutionary technology that has attracted the attention of researchers and innovators in the technical field. We also gave different definitions of this new invention and concluded that there is still no consensus. Then, we presented a basic explanation of its structure.

CHAPTER 3

DESIGN AND EXPPERIMENTATION

3.1 Introduction

In the previous chapter, we have given an overview of genetic algorithms, blockchain and the key cryptographic functions used by the blockchain. In this chapter we will show how we applied genetic algorithm and the blockchain technology to create a secure intelligent traffic light system, represent the details of the development of the genetic algorithm seen in the previous chapter. We describe the implementation of our algorithms by justifying certain technical choices such as the programming environment used, also the structure of the system and explaining the role of the classes implementing the different phases of the genetic calculation process.

3.2 Conception

3.2.1 Formalization Of The Traffic Light Problem

The first functionality of our developed system is to make a decision by the determination of the green light time allowing to minimize congestion and traffic jam flow. This calculation is provided by the use of genetic algorithm which receives inputs from the detection system. This later includs sensors and image video detection system, for example.

Vehicle detection and counting is important in the calculation of traffic congestion. It represents the input of the traffic light management system. Objects here are defined as vehicles moving or stopping on the roads. In each road there are two lines. The number of vehicles on each side and each line of the intersection is counted when the traffic light is red.

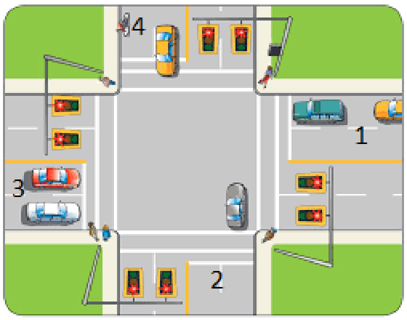


Figure 3.1: Schema of one intersection.

In this work, we consider a combination of four intersections. Thus, the system we developed will study the impact of an intersection on the two other intersections related to it. Thus, the following notations are used:

- The intersection is supposed to be the label of four roads following numbers 1, 2, 3 and 4.
- The road is supposed to be the label of two lines 1 and 2.
- All vehicles are travelling at the same speed of 20 km/h.
- In the following figure, each of the intersections 1, 2, 3 and 4, is controlled by an intelligent traffic light.
- Each two opposite lines have the same light and same time light (for example in Figure 3.1, lines 2 in road 1 and 2 in road 3 are opposite lines).
- The lines on which the vehicles will continue to drive straight or they will turn right are numbered by 1.
- The lines that have vehicles turning to the left are numbered by 2.

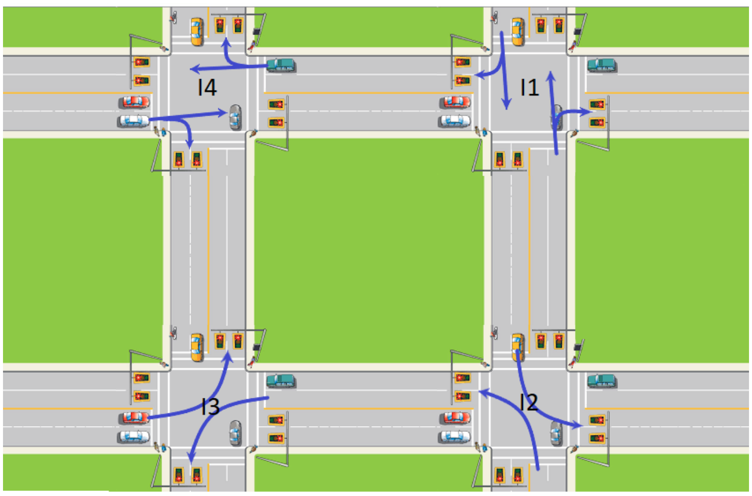


Figure 3.2: Schema of four intersections.

Note that each of the four intersections has an ITL based on GAs (first functionality of our system), so the second functionality of our system is to ensure the communication of the solution computed by an ITL located on a given intersection to the other three intersections for the purpose of verifying the adequacy of the solution to their states. However, since this communicated information may or may not be prone to malicious modifications, it must be routed in a secure way.
To apply this, we have used blockchain technology.

3.2.2 Secured Genetic Traffic Light

First, we will detail the use of genetic algorithms for the development of our proposed ITL. Then, we will explain the application of the blockchain to secure the sharing of the calculated genetic solution.

3.2.2.1 Genetic traffic light

1. **Coding:** The coding step is one of the most important steps in the genetic algorithm. A solution (chromosome) is a vector of 32 colomn (genes). Each gene includes the following information: intersection, road, line and green light time which is a random number between 10 and 45 seconds and its opposite road. As we seek to optimize or adapt the green time granted to a road according to its state, the binary coding is the most appropriate because it generates a new time value for each iteration (crossover/ mutation) of GA.

gl 1.1.1	gl 1.1.2	. . .	gl 2.4.2	gl 3.1.1	. . .	gl 4.4.1	gl 4.4.2

Figure 3.3: encoding of chromosome.

While:

- $gl_{i.j.k}$: is the time of the green lights of intersection i on road j and line k.
- 10 second $< gl_{i.j.k} <$ 45 second, and the 0 value represents the red light.

2. **Creation of initial population:** After the coding step, we will move to the next step which is the creation of the initial population. In this step we are going to create a start (initial) population composed of many individuals. Each individual is created by selecting three bits randomly from a random gene then applying the XOR operator on the selected bits.

Algorithm: Creation of initial population(initial_ch)

```
For I = 1 to pop_size do
    For x in ch_size, step = 8 do          # for each gene in the chromosome
        C_gene ← random(x, x+3)            # select a random position 'gene'
        Convert int_time(gen[c_gene]) to bi_time   # convert integer time to binary time of the selected gene
        (b1, b2, b3) ← random(2, 7)        # select 3 random positions
        Switch bi_time[b1, b2, b3]         # apply XOR in the selected positions, 0 → 1 or 1 → 0
        Ind ← initial_ch                   # initial_ch is the initial chromosome
        C_gene_pair ← pair(gene)           # pair(gene) is the gene's peer
        Int_time(gene[c_gene]) ← Reconvert bi_time to int_time
        Int_time(gene[c_gene_pair]) ← Reconvert bi_time to int_time
        For I ← x to x+7 do                # for each gene in the current chromosome (Ind)
            If I != (c_gene or c_gene_pair) then   # if it's not the current gene we're working on it
                Int_time(ind(gene[i])) ← 0   # set the green light time to zero
            End if
            add ind to initial_pop         # add the current chromosome to the initial population
        end for
    end for
end for
```

3. **Evaluation:** In this step, we are going to give each individual (chromosome) according to a function a weight that indicates if it is ''good" or ''bad".
The fitness function that we have chosen to evaluate individual forming a certain population is the following:

$$W = 50 - Td$$

W: is the weight that we give to each individual.
Td: is the difference of time between individual time and the calculated time. It is calculated by the following function:

$$Td = |It - Ct|$$

It: is the Individual time which represents the green time.
Ct: is the calculated time which is calculated from the current stat of the road by the following function:

$$Ct = Vn * Mt + Crt$$

Vn: is the vehicle number in the selected line.
Mt: is the moving time that a vehicle needs to move to the next position.
Crt: is the crossing time that a vehicle needs to cross the intersection.

$$Mt = \frac{S}{Cl} \qquad ; \qquad Crt = \frac{S}{Cd}$$

S: is the vehicle speed = 20km/s.
Cl: is the (car length) = 5m.
Cd: is the crossing distance = 25m .

4. **Selection:** After giving each individual (chromosome) its weight, we classify the population according to their weight. Then, we select the best individuals to be the parents on the next generation, using the wheel selection method [26].
The wheel method principle is the same as that of casino roulette. It amounts to imagining a kind of casino roulette on which are placed all chromosomes of the population according to their fitness values, then the ball is thrown and stops on a chromosome.
By using the wheel method, the individual which has the better weight is the one which has more chances to be selected.

Algorithm: selection(pop, n)

```
1: New_pop[]                                   # create empty list for the new population
2: W_sm ← calculate_weight_sum(pop)
3: For I ← 0 to n do
4:        Ch ← roulette_weel(pop, w_sum)  # select a chromosome using roulette wheel method
5:        Add ch to new_pop               # add the selected chromosome to the new population
6:        Remove ch from pop              # remove the selected chromosome from the current population
7: End for
```

Algorithm: calculate_weight_sum (pop)

```
1: S ← 0
2: For ch in pop do          # for each chromosome in the population
3:     S ← s + weight(ch)    # weight(ch) is the chromosome's weight calculated by the fitness function
4: End for
```

Algorithm: roulette_wheel(pop, w_sum)

```
r ← random(0, w_sum)        # w_sum is the weight's sum of the chromosomes in the population
p ← 0
For ch in pop do            # for each chromosome in the population
    p ← p + weight(ch)
    If p >= r then
        Return ch           # return the chromosome (selected by the roulette wheel)
    End if
End for
```

5. **Crossover:** Crossover step can create a new generation by obtaining new individuals from the individuals existing in the population, which improve the quality of the population.
   The crossover operation chooses randomly two individuals from the old generation and applies the cross on them. The result will be two new individuals (if the two new individuals are identical, the result is just one individual)
   The new generation will contain the old individuals (parents) and the new individuals (children). The crossover operator that we applied here is the one-point crossover. One-point crossover steps:

   1. Chose randomly two individuals (parents) from the population.
   2. Select randomly a crossover point (8, 16, and 24).
   3. Copy the first sequence of Parent-1 into the descendant Child-1 at the same positions.
   4. Complete the rest of the Child-1 chromosome the second sequence of Parent-2.
   5. Do the same with Child-2.
   6. In the first part of the chromosome we choose a randomly gene that its gene's green time doesn't equal 0 in the first child or the second one.
   7. Suppose that the binary time of the gene selected as a chromosome.
   8. Do the crossover steps on it.
   9. Add all of Child-1, Child-2, Parent-1 and Parent-2 to the new population. And remove it from the current population.

Algorithm: Crossover(ch1, ch2)

```
1:  pos ← random.choice(8, 16, 24)                  # select a random position (8 or 16 or24)
2:  child1 ← [ch1[i] for I ← 0 to pos] + [ch2[i] fori ← pos to ch_size]   # create
        chromosome child1 and child2 using crossover between parent1 and parent 2
3:  child2 ← [ch2[i] for I ← 0 to pos] + [ch1[i] fori ← pos to ch_size]   # ch1 &
        ch2 are parent1 & parent2
4:  b ← random(2, 7)
5:  while true
6:      pos11 ← random(0, 7)
7:      if(int_time(gene[pos11](child-1)) != 0)     # if the green light time of the selected position
                                                      (gene) not equals zero
8:          break
9:  while true
10:     pos12 ← random(0, 7)
11:     if(int_time(gene[pos12](child-2)) != 0)     # if the green light time of the selected position
                                                      (gene) not equals zero
12:         break
13: bi_time_ch1 ← Convert child1(int_time(gene[pos11])) to bi_time
14: bi_time_ch2 ← Convert child2(int_time(gene[pos]12)) to bi_time
15: child1 ← [bi_time_ch1 [i] for I ← 0 to b] + [bi_time_ch2 [i] fori ← b to 8]
16: child2 ← [bi_time_ch2 [i] for I ← 0 to b] + [bi_time_ch1 [i] fori ← b to 8]
17: int_time(gene[pos](child-1) ← Reconvert bi_time to int_time
18: int_time(gene[pos](child-2) ← Reconvert bi_time to int_time
19: repeat 5-18 (pos21, pos22 ← random(8, 15)      # repeat steps(5 – 19) for ( random(8,15))
20: repeat 5-18 (pos31, pos32 ← random(16, 23)     # repeat steps(5 – 19) for ( random(16,23))
21: repeat 5-18 (pos41, pos42 ← random(24, 31)     # repeat steps(5 – 19) for ( random(24, 31))
22: do mutation(child1)                             # do mutation for child1
23: do mutation(child2)                             # do mutation for child2
```

6. **Mutation:** To accomplish the mutation operation of our study, an individual will be selected randomly, then a gene will be randomly selected too; on which a permutation is done. The resulting individual will be added to the population.

Algorithm: mutation(ch)

```
 1: p ← random(0, 1) / (step=0.01)
 2: If p<p_mut then                          # p_mut is the mutation rate
 3:     while true
 4:         pos ← random(0, ch_size)         # selecting a random position(a gene) in the chromosome
 5:         if(int_time(gene[pos](ch)) != 0) # if the green light time stored in the gene doesn't
                                               equals zero
 6:             break
 7:     b ← random(2, 7)
 8:     Convert int_time(gene[pos]) to bi_time   # convert integer time to binary time
 9:     Switch bi_time[b]                    # apply XOR in the selected position 'b', 0 -> 1 or 1 -> 0
10:     Reconvert bi_time to int_time        # convert binary time to integer time
11:     gene[pos] ← int_time                 # restore the new green light time in the current gene
12:     pair(gene[pos]) ← int_time           # do the same for the gene's peer (as in 11)
13: endif
```

7. **Stopping criteria:** The execution of the previous steps (evaluation, selection, crossover, mutation) will be completed when the iteration number reaches the specified number (the stopping criteria).
   When the stopping criterion is met, we are going to select the fittest chromosome which is the best solution from the current population.
   However, since we are working on a traffic light system (four intersections), we have to protect it from any attacks and build a secure one. That will be the main reason to apply the blockchain technology in this work.

3.2.2.2 GTL secured by blockchain

Let us recall that our work includes proposing an intelligent traffic light that can optimize the combined state of four intersections. The tool used is genetic algorithm, and its application details are as described above. The developed GTL proposes green time plan traffic controllers for all four intersections.
We describe below the outline of the process used to securely verify and distribute the AG-computed combination of the next traffic light state.
Since each intersection is considered a client in the intersection network, the following senario summarizes the main verification and secure "negotiation" steps of the solution:

1. At some point, the intelligent calculation of the next green time schedule will be launched on all four GAs corresponding to the four intersections.

2. Each of the four combinations proposed as possible solutions to the problem will represent the corresponding client intersection transaction.

3. The four transactions will be sent for verification and secure distribution to the blockchain (all miners). An important point to note at this level is that we don't necessarily have to wait for the four transactions to be ready before sending them to the blockchain. If a GA located on a certain intersection converges, it will send the calculated solution (combination of schedules) to the blockchain without waiting for the convergence of the other three GAs.

4. The mining operation will be launched, where the first miner who validates his block distributes it for verification to the blockchain network.

5. After block verification, it will be dated and added to the blockchain that all users can access.

6. Finally, the four clients will receive the solution corresponding to the transaction carried by the validated block.

3.3 Experimentation And Result

3.3.1 Programming Environment

There are several programming languages that support simulation, we chose PYTHON for the following reasons:

- It is an interpreted language, making interactive exploration of code or data possible, and providing immediate feedback to the user.

- Python has strong support for task automation

- Python is a highly extensible language: means You can work with and deploy Python applications in nearly any environment, and there's little to no performance loss no matter what platform you work with.

3.3.2 Structure

In this section we present a description of the structure of the application which includes the following set of classes:

- **Class Case:** This class defines the position of the component parts on the road, such as the location and some attributes of vehicles and traffic lights.

- **Class Car:** This class contains the parameters of the vehicles, their position and movement on the different roads of the intersection.
  Some methods in class Car:

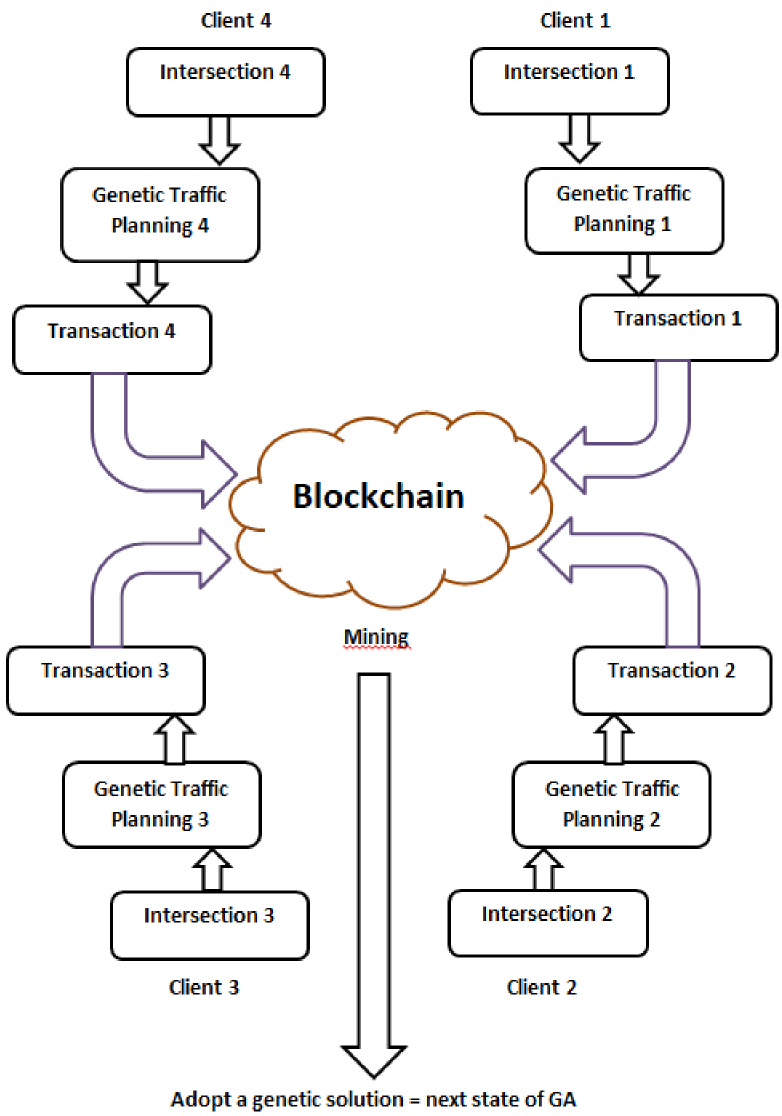


Figure 3.4: Blockchain-based security of the genetic combination calculated for the GTL.

  - **get_new_direction():** let us get the next direction of the car depending on its current state or current direction.
  - **move_car():** to move the car(object) to the new position.

- **Class Road:** Contains all the parameters and information about the vehicles in the road.

- **Class Intersection:** Contains all the information of the roads, and since we are setting up an intersection as a node minor (thread), it should contains also the population of the GA, because all calculations happen here in the intersection object.
  Methods of class intersection:

  - **Fitness (chromosome):** for weight's calculation of the chromosomes (individuals) of the populations.

  - **Crossover(chromosome1, chromosome2):** apply the crossover on two chromosomes.
  - **Mutation(chromosome):** after crossover we apply mutation on the two new children.
  - **Selection():** this method implements the application of the roulette using the evaluation scores. The individuals that will form the new population.

- **Class chromosome:** contains chromosome's information (genes, weight, size).

- **Class gene:** contains information of the gene such as the green light time, etc.

And some methods like:

- **create_initial_chromosome():** to create the initial chromosome in a random way.

- **Create_initial_population(initial chromosome):** to create the initial population starting from the initial chromosome.

3.3.3 Genetic Parameters And Illustrations

Through this section, we will present the approach followed to set the Algorithms parameters and illustrated implementation of genetic algorithms and blockchain.

3.3.3.1 Genetic parameter

After doing some trial executions, we found the following parameter for each of chromosome size, population size (individual number), generation number, selection number, mutation probability.

Population size	Generation number	Selection number	Mutation probability
100	15	250	0.07

Table 3.1: Genetic parameter.

3.3.3.2 Illustration

- **Chromosome coding:** As we mentioned in the previous chapter, the chromosome is represented by 32 cases (gene). A chromosome represents the information of four intersections, each one contains four roads, and a road has two lines. The following figure explains the chromosome coding:

3.3.3.3 Generation of the initial population

The figure below shows two examples of chromosomes generated from the initial chromosome shown in the previous section.

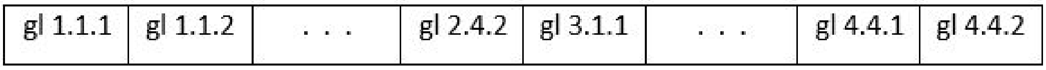


Figure 3.5: chromosomes generated from the initial chromosome.

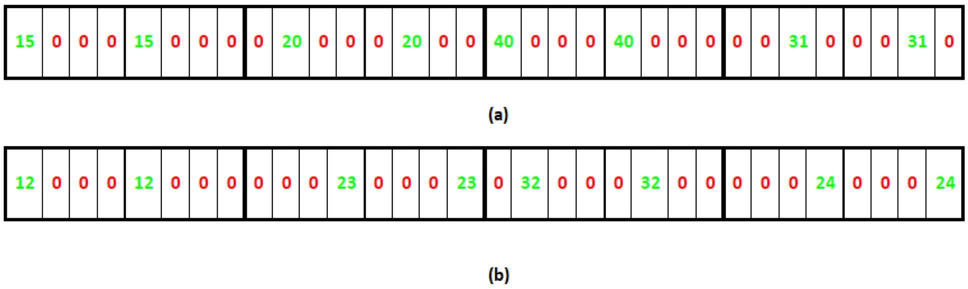
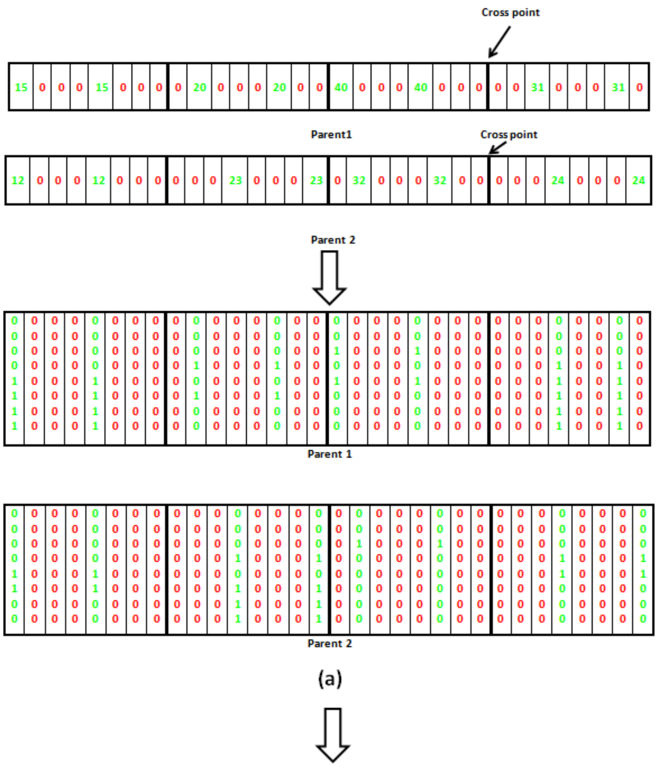


Figure 3.6: chromosomes generated from the initial chromosome.

3.3.3.3.1 Reproduction operators

- **Crossover:**
  The crossover method we have chosen is one-point crossover that allowed to choose a randomly cross point and create new children by permuted between parents(figure (a)). After that, we worked on the second part of the chromosome, we choose the genes having green light in each chromosome (figure (b)). Then, choose a random cross point and permute between those genes (figure (c)). Finally, add the two new children and the parents to create the new generation.



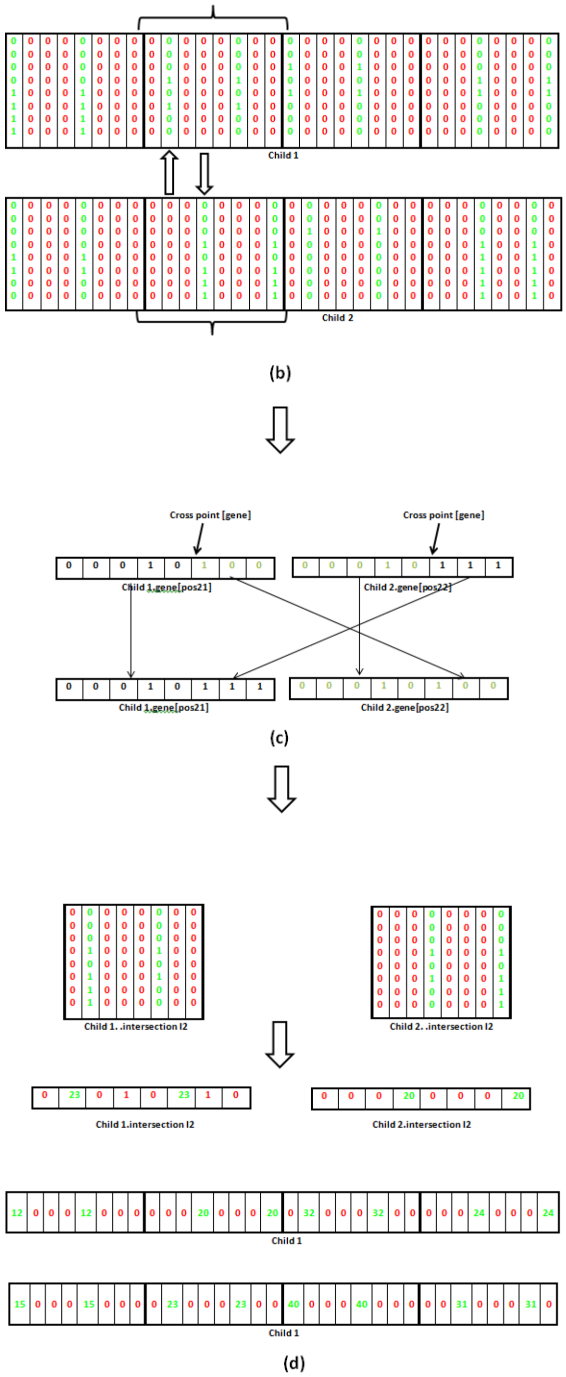


Figure 3.7: Example for one-point crossover.

- **Mutation:**
  The following figures present a child have generated by the mutation operation:

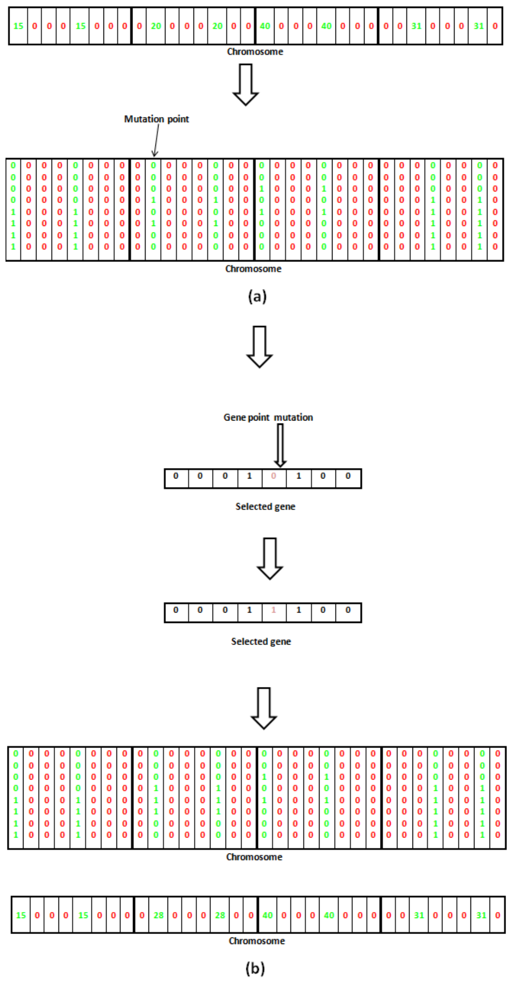


Figure 3.8: Example of mutation.

3.3.4 Results And Discussion

In the previous section, we presented examples of genetic processing of green time combinations. At this point, we will present, interpret and discuss some results obtained by our system.
The following tables illustrate three states of four intersections, specifying the number of vehicles on the different lines of the four intersections in each of the three cases.
The first two states (shown in the first two tables) are of the same "complexity" where they have the same overall number of vehicles (18 vehicles). The third state has more complexity where the total number of vehicles is 79 vehicles.

	Intersection 1		Intersection 2		Intersection 3		Intersection 4	
number of vehicles	line 1	line 2	line 1	line 2	line 1	line 2	line 1	line 2
	5	2	0	3	1	0	2	5

Table 3.2: Example 1.

	Intersection 1		Intersection 2		Intersection 3		Intersection 4	
number of vehicles	line 1	line 2	line 1	line 2	line 1	line 2	line 1	line 2
	0	4	3	4	1	6	0	0

Table 3.3: Example 2.

	Intersection 1		Intersection 2		Intersection 3		Intersection 4	
number of vehicles	line 1	line 2	line 1	line 2	line 1	line 2	line 1	line 2
	5	11	10	6	7	12	8	20

Table 3.4: Example 3.

3.3.5 Presentation Of The Interfaces

The goal of our application is to simulate a traffic light system using a genetic algorithm. Through our design of window spread to the different sides of the work and its different functionalities, we wanted an easy use for our application.

3.3.5.1 The home window

Our application based on a simple and easy to use window that allows for all kind of people to use it. The following figure represents the home window of our application.

3.3.6 Experimental Results

In this section, we report on examples of application of our system on different road conditions.
Example:

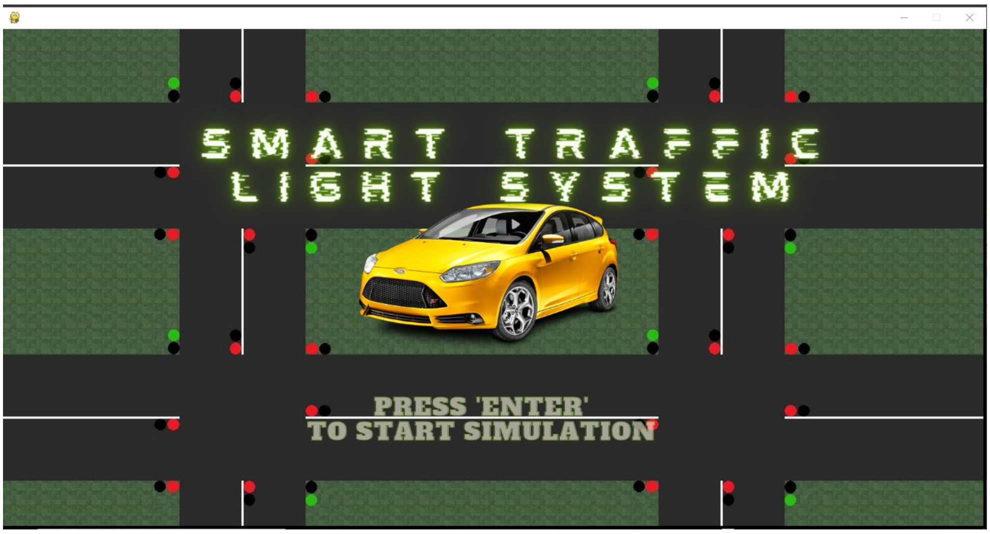


Figure 3.9: main window of our application.

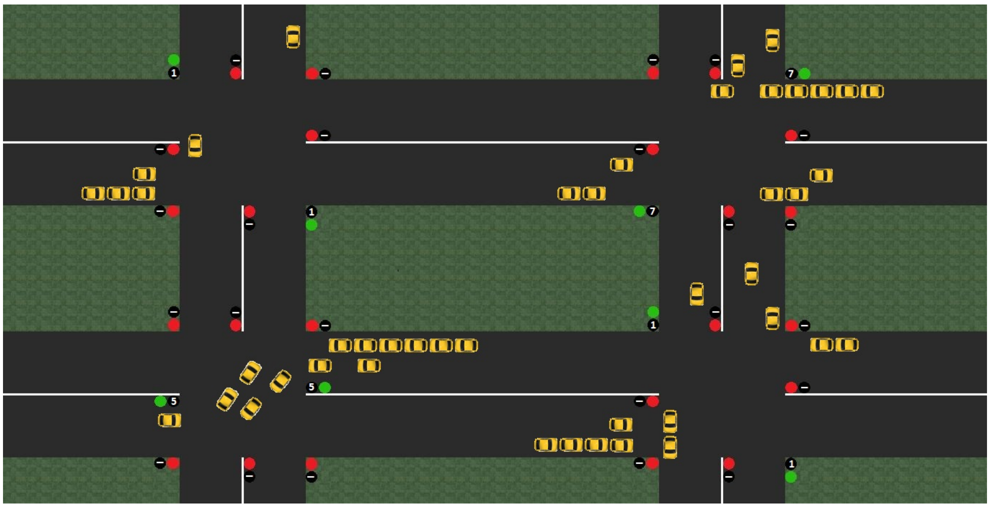


Figure 3.10: congestion example.

Consider a road condition shown in Figure 3.10, where a congestion problem occurs primarily on road 1 line 1 at intersection 3.

As shown in Figure 3.10, there is a traffic jam in intersection 3 on road 1 line 1. After applying the solution computed by our system, the road starts to clear by passing most of the vehicles that were waiting in front of the red light.

3.4 Conclusion

In this chapter, the details of the structure of this work have been introduced and explained. We have set out in detail the different stages that make up the general structure of the process developed: encoding, creation of the initial population, crossover, mutation, selection and stopping criteria.
Also, we have talked about the security in the system and how we applied the blockchain

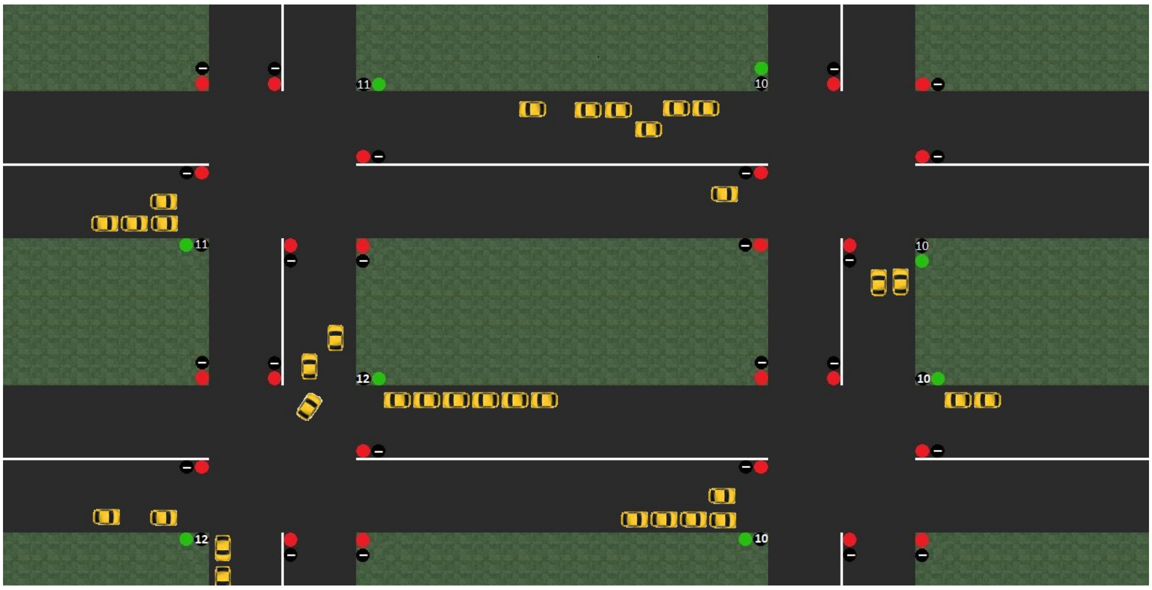


Figure 3.11: road state.

technology.
Also, we have presented the structure of the application to simulate our proposed system that we have developed using the PYTHON, The adopted parameters was also presented followed by an illustration of the main steps of the genetic process of development namely the generation of the initial population and the application of the selection, crossover and mutation.

General Conclusion

Modern traffic management methods at intersections (mainly, traffic signals as the subject of our research) have improved traffic conditions to a large extent.
The purpose of our work was to design and develop a new intelligent traffic light management system exploiting the genetic algorithms of meta-heuristic, mainly, because of its large use of randomness very promising and very interesting to exploit in such a field and giving rise, thus, to a dynamic system adaptable to changing traffic conditions.
Thus, by knowing the number of vehicles and the average waiting time at an intersection, it will be possible to estimate the green light times forming the first component corresponding to the initial chromosome from which our system calculates, over the iterations, the best combination of green times optimizing the management of road traffic at intersections.
In other words, our method uses genetic algorithm to solve the traffic light problem, which finds the best solution (time) for the green light. The proposed solution should be an optimal or a best solution for the whole city (4 connected intersections, in our work). Then, just the main lines of operation of a blockchain to verify and secure the sharing of the calculated solution between the intersections, were summarized.
As perspectives for this work, we consider:

- Simulation of the genetic process proposed for a more appropriate presentation and a more explicit highlighting of the results obtained.
- Develop in detail the application of blockchain technology as an alternative to verify and secure the sharing of the calculated combination between the different intersections.

BIBLIOGRAPHY

[1] Fabricio Alvarado Luis Jácome Leonardo Benavides Diejo Jara, Gonzalo Riofrio and Manuel Pesantez. A survey on intelligent traffic lights. 2018.

[2] Jelle Van Veenen Jilles Vreeken Marco A. Wiering and Arne Koopman. Intelligent traffic light control. 2004.

[3] Luiz Fernando Pinto de Oliveira Leandro Tiago Manera and Paulo Denis Garcez da Luz. Development of a smart traffic light control system with real-time monitoring. *JOURNAL OF LATEX CLASS FILES*, 14(8), 2015.

[4] N. Villanueva-Rosales L. Garnica-Chavira V. M. Larios, L. Gómez and E. Aceves. Semantic-enhanced living labs for better interoperability of smart cities solutions. 2016.

[5] Pan Shiyin Zhang Yongzhong, Wang Li and Li Zhengxi. Urban traffic complex network hub node analysis and signal control optimization strategy research. 2010.

[6] N. B. Hounsell F. N. McLeod and P. Burton. Scoot : a traffic database. *Third International Conference on Road Traffic Control*, pages 99, 103, 1990.

[7] Stevanovic Aleksandar Kergaye Cameron and Peter Martin. Scoot and scats: A closer look into their operations. 2009.

[8] Wei Feng Miguel Figliozzi, Courtney Slavin and Peter Koonce. Statistical study of the impact of adaptive traffic signal control on traffic and transit performance. *Transportation Research Record Journal of the Transportation Research Board*, pages 117, 126, 2013.

[9] Lin Zhang and P. D Prevedouros. Signal control for oversaturated intersections using fuzzy logic. 2008.

[10] Khattab M. Ali Alheeti. Intelligent traffic light control system based image intensity measurement. 2011.

[11] Ayad M. Turky Mohd Zaliman Mohd Yusoff and Ayad Sharifuddin Ahmad. The use of genetic algorithm for traffic light and pedestrian crossing control. *International Journal of Computer Science and Network Security*, 2009.

[12] SAJANA SAMARASINGHE. Top 10 smart cities worldwide. *businesschief*, 2021.

[13] The history and meaning of colored traffic lights. *IDRIVESAFELY*.

[14] Lise Wagner. 1868-2019: a brief history of traffic lights. *inclusivecitymaker*.

[15] M. AbuNaser A. A.Tawalbeh, A. Tamimi and K. Saleh. Intelligent traffic light based on genetic algorithm. pages 851, 854, 2009.

[16] D. Srinivasan M. C. Choy and R. L. Cheu. Neural networks for real-time traffic signal control. 7(3), 2006.

[17] Hadj Rabah Sabrina and Nouari Djahida. Feux tricolores intelligents pour les villes intelligentes. *Universit ´e Akli Mohand Oulhadj de Bouira Facult ´e des Sciences et des Sciences Appliqu ´ees D ´epartement d'Informatique*.

[18] E. Kougianos U. Choppali and Saraju P. Mohanty. Everything you wanted to know about smart cities. pages 60, 70.

[19] Intelligent traffic light control using neural network with multi-connect architecture. *Scientific Figure on ResearchGate*, 2021.

[20] Jenna Carr. An introduction to genetic algorithms. 2014.

[21] Denny Hermawanto. Genetic algorithm for solving simple mathematical equality problem. *Indonesian Institute of Sciences (LIPI) INDONESIA*, 2013.

[22] I.C.LERMAN and R.F.NGOUENET. Algorithmes génétiques séquentiels pour une représentation affine des proximités,. *Institut National de recherche en informatique et en automatique (INRIA)*.

[23] A.SOUQUET and F.G.RADET. Algorithmes génétiques. *TE de fin d'année. Université de Nice Sophia Antipolis,*.

[24] Recioui A. Use of genetic algorithms in antennas: Application to yagi-uda antenna and antenna arrays. *Mémoire de magister, Université M'Hamed bouguerra Boumerdes*, 2006.

[25] Yahia Rahmat-Samii and Eric Michielssen. Electromgnetic optimization by genetic algorithms. 1999.

[26] Coley D. , an introduction to genetic algorithms for scientists and engineers. , *Singapore: World Scientific Publishing*, 1999.

[27] C. RAMDANE M. HADJ-RACHID, C. BLOCH and P. CHATONNAY. Différentes opérateurs évolutionnaires de permutation : sélection, croisement et mutation. *Rapport de recherche, Université de Franche Comté de Montbéliard*, 2010.

[28] M. RENVERSADE. Optimisation d'un dispositif hyper-sustentateur par algorithmes génétiques. 2006.

[29] S.N.Sivanandam and S.N.Deepa. Introduction to genetic algorithms. 2008.

[30] R.Cheng M.Gen and L.Lin. Network models and optimization: Multiobjective genetic algorithms approach. 2008.

[31] M.Gen and R.Cheng. Genetic algorithms and engineering optimizations. 2000.

[32] T.Dodok N.Čuboňová and Z.Ságová. Optimisation of the machining process using genetic algorithm. *Scientific Journal of Silesian University of Technology*, 2019.

[33] Satoshi Nakamoto. Bitcoin: A peer to peer electronic cash system. 2008.

[34] Yang Chen Xinle Yang and Xiaohu Chen. Effective scheme against 51history weighted information. *MOAC Blockchain Tech Inc*, 1(1):1–19, 2019.

[35] Hongning Dai Xiangping Chen Zibin Zheng, Shaoan Xie and Huaimin Wang. An overview of blockchain technology : Architecture, consensus, and future trends. 2017.

[36] Jean-Guillaume Dumas and Pascal Lafourcade. Bitcoin: une monnaie dématérialisée. 2017.

[37] Dijana Jagodic Dejan Vujičić and Siniša Ranđić. Blockchain technology, bitcoin, and ethereum: A brief overview. 2018.

[38] Gautam Dhameja Bikramaditya Singhal and Priyansu Sekhar Panda. Beginning blockchain. 2018.

[39] Imran Bashir. Mastering blockchain. *Packt Publishing Ltd*, 2017.

[40] Jonatan Bergquist. Blockchain technology and smart contracts. 2017.

[41] Rui Xue Rui Zhang and Ling Liu. . security and privacy on blockchain. *ACM Computing Surveys (CSUR)*, 2019.

[42] Blockchain consensus algorithms and mechanisms : Startup guide for beginners.

[43] Sunny King and Scott Nadal. Ppcoin: peer-to-peer crypto-currency with proof-of-stake. 2012.

[44] Sigrid Seibold and George Samman. Consensus : Immutable agreement for the internet of value. 2016.

[45] H. B. Amnieh D. J. Armaghani R. S. Faradonbeh, M. Hasanipanah and M. Monjezi. Development of gp and gep models to estimate an environmental issue induced by blasting operation. 2018.

[46] J. M. Johnson and Y. Rahmat-Sami. Genetic algorithms in engineering electromagnetics. 1997.

[47] Laxmi Sharma Rahul Nyamangoudar Abhishek Javali Sudhir K. Routray, Mahesh K. Jha and Sutapa Sarkar. Quantum cryptography for iot: Aperspective. 2017.

[48] K. Rajeshwaran and K. Anil Kumar. Cellular automata based hashing algorithm (cabha) for strong cryptographic hash function. 2019.

[49] Dwi Liestyowati. Public key cryptography. *J. Phys.: Conf. Ser. 1477 052062*, 2020.

[50] Mohammed.F Alhamid zulfiqar Ali, Ghulam.Mu. An automatic health monitoring system for patients suffering from voice complication in smart cities. Mar 2017.