An Internet of Things (IOT) based Smart Parking Routing System for Smart Cities

Elgarej Mouhcine[1], El Fazazi Hanaa[3],
Khalifa Mansouri[4], Youssfi Mohamed[5]
Laboratory SSDIA
ENSET University Hassan II
Mohammedia, Morocco

Karouani Yassine[2]
Laboratory RITM
ENSEM-ESTC-UH2C
Casablanca, Morocco

Abstract—**Recently, the number of cars on the road has been growing due to the increase in car manufacturing in parallel with customer services provided to help the new driver to buy cars at affordable prices. On the other hand, we find that the infrastructure of big cities cannot support this number of cars and with the disorganization of parking places in the city this problem will lead us to have serious problem in the city which involves the increase of drivers requests to find the nearest parking places to avoid traffic congestion in those areas. In parallel today we talk about the concept of smart cities and how can we use the evolution of the Internet of Things (IoT) to improve the quality of the smart city. Several efforts have been made on the Internet of Things to improve the reliability and the productivity of public infrastructure. Many problems have been handled and controlled by the IoT such as vehicle traffic congestion, road safety and the inefficient use of car parking spaces. This work introduces a novel technique based on a distributed cloud architecture of IoT to manage the parking systems combined with a distributed swarm intelligence technique using the Ant System algorithm to improve the process of finding the nearest car parking in the minimum time based on the state of traffic on this road. This prototype will help drivers to find the nearest car parking and improve the exploitation of the available car parking in the city.**

Keywords—*Intelligent traffic system; internet of things; swarm intelligent; ant colony optimization; vehicle routing system; multi-agent system; smart parking system; smart cities; cloud computing system*

I. INTRODUCTION

The theory of internet of things (IoT) began with objects as a pre-identified devices. Those devices can be controlled, tracked and monitored by using a remote system connected through the internet. IoT is based on the internet that will be used as a link of communication between things and physical objects (devices) through a network of communication. This new concept is based on two words "internet" and "things", the first element presents the most tool of communication between several entities (computers, phones, tablets, servers, etc.) based on a set of protocols of communications to achieve the processes of (in/out) of data between several entities. The second word "Things" presents a physical or logical devices or objects viewed as a final device on this network system. IoT, in general, presents a set of devices used for data transfer between a set of entities toward organizing and managing units to analyze the received data. These things are combined with embedded devices that allow them to compute and communicate with the available devices in the network.

Actually, a city can be viewed as a smart city [8], [19], [20] by making the majority of his areas smart, such as water supply distribution, electricity distribution policy, smart transport distribution [11], [23] and smart health-care vehicles. Making the majority of these entities smart will lead us to build a smart city. Vehicle traffic control [1], [14] is one of the famous problems faced by each city. Hence, several works have been done to control and manage traffic jam based on traffic signal control [2], smart parking distribution system [4], [17] and smart route navigation [6]. The idea of building a smart city is now possible with the evolution of the internet of things and one of the main issues of the smart city is to find a new concept to control traffic road [2], [22] and managing car parking places [5]. Recently, in big cities, the object of finding available parking space [18] is become a serious problem and even is become harder with the increase in the number of vehicles on the road. In order to solve this problem, the new smart cities should take this problem into consideration which leads us to decrease the searching time for parking slots and decrease the traffic flow on the route. In the case of parking systems, the driver needs some smart solutions that inform them about the available car parking around them based on the available embedded devices. Many modern cities have decided to implement a set of sensors and devices technologies based on the concept of IoT to control and manage all the traffic data on the city. Actually, several systems have been made for car parking system that allow the driver to receive real-time information about the state of parking slots and the available parking on a specific area in the city only based on the web pages or using mobile applications. Each parking, use a set of sensors to control and manage the state of parking slots.

In this wrok, a new concept of smart parking system will be developed to reduce the time consumed by the driver to find an available parking space to park his vehicle also to reduce the traffic flow in the city [18]. Based on such a system, the user will be able to find the nearest parking spaces and they will propose for them the shortest path (in terms of distance and road traffic) to arrive at the location of the parking. In this work, a distributed solution will be introduced to find the available parking slots for a driver based on the technology of IoT and propose to him the shortest path in terms of the distance and the traffic flow to arrive in the best conditions based on the technique of the ant colony optimization [26]. The system allows users to choose the optimal route to reach their destination by looking for the nearest car parking around

of them.

The rest of this paper is organized as follows: Section II looks at previous works which are related to the parking routing system, Section III describes the IOT concept and background. Section IV formalizes the problem of parking routing system based on the services provided by the IOT technologies. The ACO concepts are described in Section V. The distributed solution for the parking system based on the technologies of IOT and the AS optimization method is shown in Section VI. Section VII looks at on our simulation results based on a multi-agent environment for the parking system (PS), the paper ends with a conclusion and offers some research direction for the future.

II. RELATED WORK

In this work [5], the authors propose a new advanced driver assistance systems (ADAs) which aim to detect vacant parking slots, then the selected parking will be recommended to this driver. The proposed system will start searching for the available parking slots and help the driver to arrive at this parking based on a guiding monitoring system dedicated to this task. When the driver arrives at the parking, the system will inform the others parking with the state of the newly available slots on this parking.

Other solutions have been made to solve the traffic jam problems by using the internet of vehicles combined with the cloud computing and the big data to control and manage the traffic routing system and made some optimal paths for vehicles and guiding them toward the available parking. In this work [24] a novel approach for routing vehicles to reach their destinations including parking spaces in the minimum time. The proposed system provides an intelligent travel guiding system that helps the driver in the urban area and avoid routes with a higher level of traffic congestion.

A genetic algorithm (GA) has been implemented in the parking routing system to provide a new strategy based on the behaviour of the GA to solve the problem of the parking systems. Authors in paper [25], create a new intelligent solution to find a series of best spaces of the moving cars at a parking space. With this method, they can route the vehicle inside the parking to achieve the best free slot in the minimum time.

The object of this paper [2] is to propose a new model that implement traffic system things to manage and control the state of the road network remotely. They propose a smart adaptive traffic system connected to a set of things using the internet which allows them to retrieve real-time information at each moment. The data collected from different components are controlled and analyzed by the traffic control office. The obtained data will allow as to evaluate the state of the road network at a specific area. Also, this system is able to help drivers by giving them real-time information about the traffic flow around this area.

In this work [3], authors try to give us the basic route to take into consideration when we implement the services of IoT, because users can access to anything only by using the Internet services over lightweight devices, also know all services are shared between user based on the new concept of cloud computing in which knowledge is moved away from

hard devices like (Personal computer or mobile phones) to enormous computers named as datacenters. This architecture will help us to improve the quality of our services by allowing all users to access to data using the internet and the newly provided services are no more centralized because in CC we talk about distributing data between users using internet services.

III. IOT LITERATURE AND BACKGROUND REVIEW

The technology of IoT is playing an important role in the network of smart technologies, so it should be able to ensure a continuous connection between unlimited smart things to the internet. Based on such technology, we need to define a standard template for the IoT architecture that will be used to prepare a smart environment based on the technologies of IoT. In the architecture (Fig. 1), we have three levels defined on this system: Presentation, Middleware, Hardware level. Different architectures have been proposed in this context but all of them are based on these three layers. The Hardware level control and manage the set of physical sensors and embedded objects, the second level propose the set of components that will be used for the cloud computing and finally, the presentation level, used for proposing a visible vision for the final user based on some human interfaces, etc.
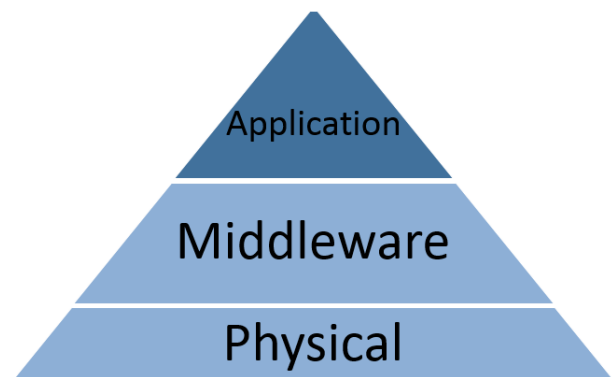

Fig. 1. The standard IoT architecture

A. Physical Level

The physical layer is based on a set of physical sensors and objects and the data received from those entities are collected to be analyzed and used for managing those entities. All the received information will be forwarded toward the middleware level using a secure channel of communication. Finally, all the data collected from those objects are stored and collected at this physical level.

B. Middleware Level

In this level, we will forward the data collected from the higher layer to the cloud system, we can use several technologies to transfer information such as 4G GSM, Wi-Fi, ZigBee, Ethernet, Bluetooth, etc. All the process is secured using TCP (Transmission Control Protocol) protocols and the data are transferred using HTTP (Hypertext Transfer Protocol) or FTP (File Transfer Protocol) mechanism. On the other hand,

this layer has the ability to retrieve data from the cloud and help the physical layer in step of collecting the data. Based on this level, we can develop a set of applications which implement and re-use the received data to propose several solutions only based on the interconnected objects inside this platform.

C. Application Level

This level proposes a set of services for customers, it handles all the requests and the responses according to the parameters set for each request. We know that our architecture is based on cloud IoT system, so according to this request, this layer can search and find the set of objects that can give us such kind of information. This one is the main interface between human and cloud IoT. Actually, we see the combination of IoT with cloud computing to improve the quality of services provided by both of them. The IoT can improve his technologies such as the links of communication and storage process, energy consumption by implementing the unlimited resources of cloud computing. Basically, the platforms of cloud computing dedicated to the IoT technologies can be viewed as an intermediate actors between applications and things in the IoT network, in order to reduce all the functionalities and the complexities for preparing a good environment to run these applications. Several factors will push us to think about the strategy of combining the IoT with cloud computing such as:

- The Availability: with the new improvement on the cloud, the availability of resources are an important element especially with the cloud integration which aims to keep alive the set of resources running and up on time based on some services built for this job.

- Processing power: in the IoT environment we are using a set of sensors with limited processing capacity and when the sensor collect and capture the data it should be transferred to more powerful devices and objects where we can apply the rest of processing to retrieve the information needed from these things. With the help of cloud technologies, we can benefit from the real-time and the parallel processing performed by the cloud network.

- Middleware services: based on the cloud computing we can use the set of the link of communication used to transfer and tracking and managing objects from any where only using the internet (such as, 4G, Ethernet, Wi-Fi, etc.). With this remote technology, IoT can control and communicate with any types of things using remote communication.

- Storage limit: In the IoT architecture, we have a set of data collected from several things on the environment, we know that each thing has a limited storage capacity so the data collected or received by those devices should be transferred to the main agent that collect this information from various objects. then, the storage services provided by the cloud can help the IoT object to share and visualize the data from any places through specific web services.

IV. OVERVIEW OF THE SYSTEM ARCHITECTURE

The overview of the problem of parking routing system [18] can be described as showing in (Fig. 2), based on this architecture we can see that our system combines several entities which are distributed on the whole infrastructure and interconnected with each other based on the cloud computing platform. Some of these elements are viewed as intelligent components which are able to improve the quality of the collected data and send it to the next agent that will use this input data and propose a set of solutions for the final users. In the beginning, we need to identify all the set of entities that will work in our architecture such as parking places, road sensors, traffic road, road network, etc. Then we need to evaluate the state of the traffic road based on the set of pre-installed RSU (Road Sensor's Units) combined with the Google Traffic Services that allow us to retrieve the traffic flow on a specific area. In parallel with this process, we will find the optimal path [26] that will be followed by the driver to arrive at the nearest parking place to his location.
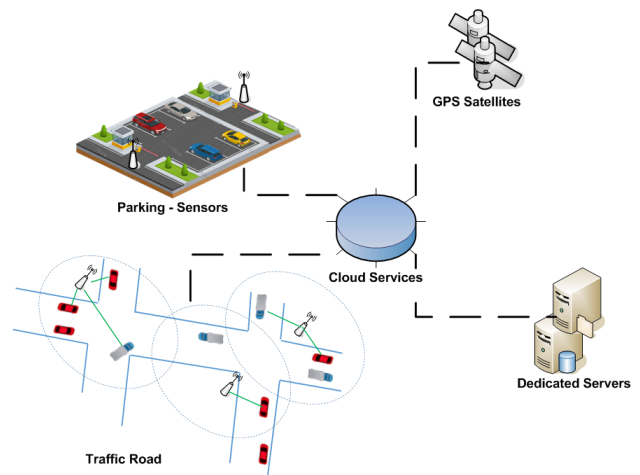

Fig. 2. Decentralized architecture for Managing the traffic road and parking system in a Smart City

A. Preparation of the Infrastructure

As shown in (Fig. 2), we need to locate all the available parking places and try to keep an eye on the state of available places at each parking, this task is done based on a set of sensors pre-installed on each parking which allow us to compute the number of available places on each parking, this collected data will be shared with a smart platform deployed on a distributed cloud system which allows us to centralize all these received data, we can identify the set of parking places on a specific area based on the classical clustering mechanism to select the set of nearest parking places to the position of the driver. In our case, each parking is controlled by an intelligent agent, this one is able to share real-update data about the state of slots on this parking at each moment. The cloud agent is able to communicate with these agents using TCP/IP network communication, based on this technology we can say that our infrastructure is remotely distributed.

B. Cloud Parking Routing System

When the cloud agent receives a request from the driver that search for a nearest parking slot, the system will catch the GPS coordinates of the driver and create an area around this position (Fig. 3), based on this area we can define the set of nearest parking places that are inside this area (we are based on the linear distance between the position of the driver and the pre-stored position of parking places in this area). When the set of available parking slots are defined, the system will prepare the path planning that will be followed by the driver toward one of those parking based on the state of the traffic road in this area and the total distance of the trip between the driver and the parking position. To get the state of traffic on this area, we are based on the RSU that allow us to retrieve the number of vehicles at a specific route and we call the Google Traffic services that give us a road network combined with traffic road. In this step, we can say that our road network for this specific area around the driver is completely prepared, then we will be able to call our distributed strategy to find the optimal path between the location of the driver and the set of parking places, in our problem the best path is viewed in terms of two elements: the total distance of the trip and the total time consumed to arrive at the location of the parking which is related to the traffic flow on this path. At the end of this process, our system will allow us to choose between a set of alternatives routes, each route will lead us to a specific parking in this area.
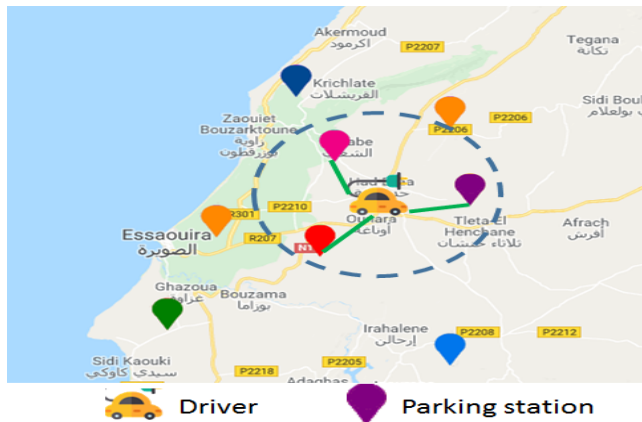

Fig. 3. Cloud Parking Clustering

V. THE PRINCIPLE OF BASIC ANT COLONY OPTIMIZATION

The ant colony optimization (ACO) technique is inspired by the behaviour of real ants in searching for food sources. Ants are initially started their journey by finding food sources randomly. If an ant finds a food source, it will leave a chemical substance named pheromone on the route used toward the source food when it returns back to their nest. It can appear that multiple ants find the same food source but with several trails. The amount of pheromone on routes will attract ants to use this trail. However, the best path or the shortest path will be more selected by the majority of ants and the amount of pheromone on this path will increase according to the number of ants that use this path. As time passes, we will see the

majority of ants will take the same path to the food source which is considered as the optimal route to the given food.

The main objective of the ACO algorithms is to create a set of artificial ants inspired by the behaviour of real ants, they can build and find optimal solutions for combinatorial problems [9], [26], and it was used to solve complex problems [10] such as the travelling salesman problem (TSP), the vehicle routing problem (VRP). The artificial agents (ants) are created to find the best paths or optimal solutions. The process is based on sequential iterations to improve the quality of the proposed solution, also it takes into consideration the previous solutions to help agents to find optimal solutions in the next iteration. Many pieces of information are shared between ants by the proposed artificial pheromone, at each iteration a new amount of pheromone is updated according to the quality of each created solution. The algorithm monitoring the state of pheromone generated by each agent according to the goodness of the solutions. On the other hand, the amount of pheromone will decrease according to the evaporation process in order to avoid convergence to local optimum solutions. The system will repeat the same behaviour until the stop condition is met.

In the real world, ants are able to find the shortest path between their nest and food sources [21] by exploring the environment based on a chemical substance called pheromone, which is released by ants on the routes they travel. The proposed Ant System (AS) algorithm [15] is a procedure used to simulate the behaviour of real ants by producing a new mechanism for artificial ants that produces the same behaviour of real ants, with the addition of several attributes that helps the system to converge toward optimal solutions in minimum time. The different ACO algorithms depend on the nature of the problem and the solution for the problem [27], so the parameters of each algorithm are based on what the user needs to search for and how the nature of the environment will be used by applying the technique of the ACO. The AS is based on a set of predefined steps defined as follows (Algorithm 1):

Graph description: artificial entities will move between a set of discrete nodes in a discrete space. Since the problem solved by the AS algorithm is viewed as a discrete problem, they can be modelled by a graph built from nodes and edges. *Artificial ants preparation:* a set of ants are distributed on the declared nodes in the proposed graph, the number of ants increases the collection of the generated solution since each ant is viewed as a candidate for the given problem. *Ants probability transition:* ants are based on a probabilistic transition to move between vertex which is based on two main parameters, which includes the pheromone and a heuristic approach that defines the nature of the optimized problem.

The AS pseudocode paremeter desrciption:

- N_{nodes}: list of available nodes on the map.

- R_{routes}: list of available routes on the map.

- A_{ants}: number of ants used to find the bests solutions.

- $best_{route}$: contain the best solution computed after all ants finished their jobs.

Algorithm 1: AS pseudocode algorithme

Input: $N_{nodes}, R_{routes}, A_{ants}$
Result: $best_{route}$

1 foreach A_i *in* A_{ants} **do**
2   **foreach** N_i *in* N_{nodes} **do**
      `// Apply a probabilistic`
      `transition based on pheromone`
      `and heuristic parameter to`
      `select best next move and add`
      `this node to CR: collection`
      `of node`
3 $n = select\_next\_node()$;
4 $C_n = n$;
5   **end**
   `// Apply a local update of`
   `pheromone to the path visited`
   `by this ant`
6 $update\_pheromone(C_n)$;
   `// Add constructed path to list of`
   `proposed routes`
7 $best_{route} = C_n$;
8 end

- $select\_next\_node()$: this function implement the equation 1 to find the next node to visited (j) when the ant is on node i.

- C_n : represents the set of already visited nodes by the given ant.

- $update\_pheromone(C_n)$: apply the function 2 for updating the value of pheromone on the set of visited nodes and routes.

At the beginning of the search process, a consistent amount of pheromone is assigned to all arcs and the cost assigned to each arc can be defined according to the nature of the problem. For each ant in the colony, it will start its own trip by visiting the set of available nodes in this graph. Node selection will be based on a probabilistic transition by using the amount of pheromone and heuristic data according to the function $selec\_next\_node()$. The visited nodes will be appended to the trip followed by the ant. When the current ant visits all existing nodes in the graph, it will then apply a local update of pheromone $update\_pheromone()$ by adding more pheromone on the set of routes visited during its search process. The current behavior will be repeated until all ants complete the existing nodes on the given graph. The system will allow us to select the best routes between the proposed solutions by ranking those solutions according to the length of each trip.

VI. THE PARKING ROUTING SYSTEM BASED ON THE IOT COMBINED WITH THE TECHNIQUE OF THE ACO

The main goal of our work is to help drivers to find the nearest parking places based on two parameters such as the distance between the current location of the vehicle and the parking slot combined with the state of the traffic on these proposed routes. To collect the state of the road we use a set of Road Sensors Units (RSU) that are able to compute the number of vehicles at a specific route and share this information with our cloud platform to be taken into consideration in the process of computing the optimal route for the driver. As shown in the previous section, the AS algorithm is based on the amount of pheromone to define the best routes visited by the majority of artificial ants. We will take the number of vehicles between two connected RSUs and correlate it with the amount of pheromone. So, if the number of vehicles is higher on this route so the amount of pheromone on this route will be the same. The number of vehicles passed between RSUs is automatically synchronized with the cloud system. This information will be used to evaluate the traffic flow between those intersections and we consider the number of stopping vehicles as the amount of pheromone on this route. This means that the pheromone density level is related to the number of non-moving vehicles on this route. The main objective of our work is to implement the behavior of the AS algorithm to search for all the best routes which have less number of vehicles (less traffic flow) and encourage the driver to go towards the route with less traffic flow. To better understand how we can implement the technique of the AS with the IOT technology [13], [16] we define the amount of the pheromone as the number of stopping cars in the line also we need to view the inverse of the original behavior of the AS algorithm by avoiding edges with higher amounts of pheromone and follow only routes with less number of vehicles (less amount of pheromone). Based on the state of the route received from the set of RSU and from the service of Google Traffic we can decide which route will be taken by the driver.

The road network will be viewed as a graph made from a set of intersections (nodes) and Routes (edges). In our case, nodes present the set of parking places and routes it will be the set of existing routes between those places. This graph will be presented as follows: $G = (I, R)$ Where I=$(I_1, I_2, ..., I_n)$ is the set of parking places and R is the set of routes. The main objective of this layout is to guide vehicles to the nearest parking in the fastest time while avoiding traffic on the road network. Avoiding traffic is done by skipping routes with a higher number of non-moving vehicles, and traffic flow delays. Each R_{ij} is identified by the distance d_{ij} and traffic flow T_{ij}. The time consumed to traverse R_{ij} is not dependent on the time consumed to traverse other routes because each route has its individual number of non-moving vehicles controlled by the RSU and the information provided by Google Traffic. The final goal is to find the optimal solution that minimizes two constraints, such as the distance and time consumed to finish the trip. So, in this problem, we will try to use two parameters (distance, traffic flow) to find the best path. The vehicle will apply a probabilistic transition to move from an intersection I_i towards I_j according to the rule below:

$$p_{ij}^m(it) = \frac{T_{ij}(it)^\alpha d_{ij}(it)^\beta}{\sum_{l \in \omega} T_{il}(it)^\alpha d_{il}(it)^\beta} \quad (1)$$

This is where the parameter (α, β) controls the influence of the traffic flow (pheromone) and the distance on the prepared solution. The parameter ω is the set of the non-visited intersections. The level of the traffic flow (pheromone) will evaporate (ρ rate of evaporation) depending on the information received from the RSU (the rate of the non-moving vehicles

on this route). The traffic flow update is done according to this equation:

$$T_{ij} = (1 - \rho).T_{ij} + \Delta_{ij} \qquad (2)$$

The argument Δ_{ij} is related to the driver to see whether the vehicle has already visited this route R_{ij} . The new amount of traffic added on this route can be computed as follows:

$$\Delta_{ij} = \sum_{n=1}^{N} \frac{R_{ij}^k}{t_m(k)} \qquad (3)$$

$tm_{(k)}$ represents the time needed to cross the distance d_{ij} of the route R_{ij} by the vehicle k based on the speed (ω_k) allowed on the route and the number of the available Non-moving cars.

$$tm_{(k)} = \frac{d_{ij}}{\omega(k)} \qquad (4)$$

So, at each intersection, a vehicle will try to select the next intersection based on (Equation 1) and will then select this route based on two parameters: the distance and the traffic flow. The system will repeat the cycle until the destination is reached or the stop condition is met. At the end, the system will show the best routes that exists between the two locations. For each route, they will give us the total travel time consumed to cross this route with the total length (km) of this trip. (Equation 1) can be transformed into:

$$p_{ij}^m(it) = \frac{T_{ij}(it)^\alpha d_{ij}(it)^{1-\alpha}}{\sum_{l \in \omega} T_{il}(it)^\alpha d_{il}(it)^{1-\alpha}} \qquad (5)$$

VII. A Distributed Strategy for the Parking Routing System

In this section, we describe the main architecture for the proposed distributed parking system which is based on several actors that collaborate together in asynchronous behaviour to build a smart parking system for vehicles in the city. The main actors that collaborate into our proposed system (Fig. 4) are a distributed agents each one of them works in a separate environment by sharing with other agents a set of parameters that are used to help drivers to find the best parking slot in the area.

A. Parking Sensors Units

In our parking infrastructure we are based on a set of sensors to control the number of vehicles (in/out) that visit our parking, also we use a set of infrared sensors under the category of Passive Infrared (PIF), those components are wirelessly connected to the Parking Agent (PA) which is modelled by a raspberry pi component using TCP/IP protocol which allows us to communicate using wifi network. So the PA will be able to communicate with those sensors and get the number of the available slots depending on the number of vehicles that leave the parking based on a simple equation by minus the number of vehicles inside the parking according to the number of vehicles that leave the parking from the outdoor sensor.
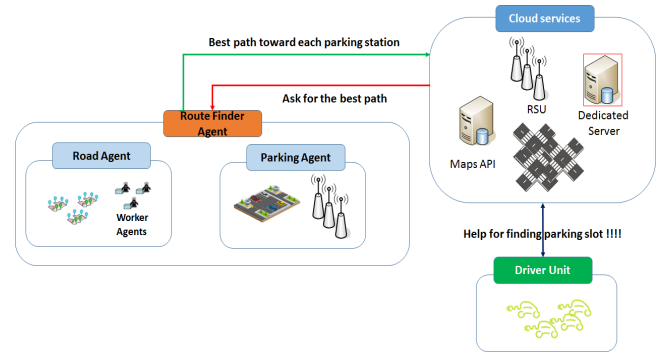

Fig. 4. Multi-agents architecture for managing the parking system based on the cloud control strategy

B. Parking Agent Unit

The Parking Agent Unit (PAU) is represented by a raspberry pi component, which is a microprocessor on a chip. This agent performs as an intermediate between the cloud routing system and sensors. All the sensors are remotely connected to the Path Finder Agent Unit. In this unit, we can find several GPIO (General Purpose Input/Output) pins, which act as an input data that allow us to connect all the available sensors in the parking. To control the process of communication between sensors and PAU we use an executable script writing with NodeJS programming language to read all the data received from the different GPIO and then transfer this data to the cloud routing system. So, we can say that our PAU will collect all data and share them with the cloud system. This data will be interpreted by other agents to select the best parking and compute the optimal path for vehicles to reach this parking.

C. Path Finder Agent and Cloud Services

In this cloud platform, we installed a dedicated server that acts as a receiver agent which collects all the data and saves it into NoSQL databases, in these databases we store all information about each parking agent such as the GPS coordinates of this parking and the real-time information about the available slots on this parking. In the other hand, we make a specific agent that creates a set of clustering area to group the nearest parking in a specific area, based on this clustering strategy the PFA can be able to fetch only the nearest area to the vehicle and optimizing the search process for the nearest parking. Also, the server can handle all the request of the users connected to our system that aims to find some available parking, according to each request, the system will invoke the PFA to respond on this demand and find an optimal solution to each request based on the available data on the server. In parallel, we define a cron job which is running at each new subscribed parking added to the list of available parking connected to our cloud system, this cron job execute a python script which create a set of clustering area each one of them contain the set of nearest parking, so at the end of this script we can see a set of groups, each one is viewed as an area that contains several parking. These areas will be used by the PFA to propose only the nearest parking to the driver and reduce the process of fetching for the nearest parking to the actual location of the driver.

As we can resume, the main function of this agent is to answer on two questions (Fig. 5) such as finding the set of parking that contains some available slots, to find the available parking we use the predefined clustering data performed by our cronjob, at the beginning the PFA will take the position of the vehicle and fetch in his local database for the nearest area to this location. Secondly, we help the driver to arrive at this parking in the minimum time and with a safety way by avoiding road with a huge traffic jam. To do this job, we are based on the technique of the AS to compute the optimal path that can be followed by the driver to arrive in the best conditions. So here we use a set of artificial ants named as worker agents (WA), each one of them will explore the graph (Fig. 6) made by our PFA, to build this graph we are based on the Google Maps API that helps us to create a map made from a set of intersection and several routes (edges) that are used to link between those intersections. To create this graph we are based on the actual location of the driver and the set of available parking places from the clustering done by our dedicated server, so based on this clustering we reduce the area that will be explored by our worker agents to find optimal routes between the driver and the set of nearest parking to the location of the vehicle. When the graph is made, each WA will try to visit the set of intersections between the location of the vehicle toward the proposed parking by the PFA, this work is done based on a probabilistic formula defined in the (Equation 1) which is based on two parameters are the linear distance and the level of traffic jam between the two adjacent intersections, when a WA visits all the available intersection it will be able to return back to the PFA with the optimal path that exists between the driver and the parking location. After all the worker agents complete their tours, the PFA will collect all the proposed paths and select the best one of them in terms of the distance and traffic jam, then this best route will be transferred to the driver to help them to arrive at this parking slot.
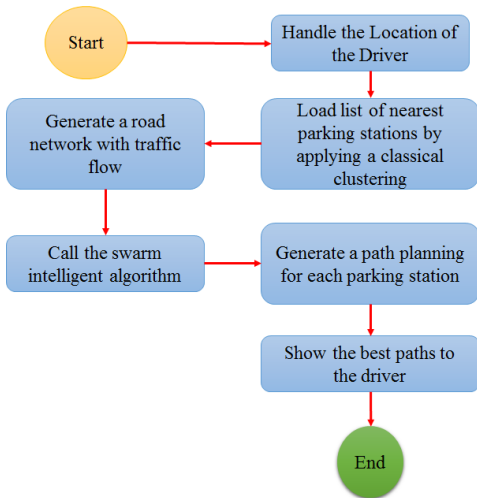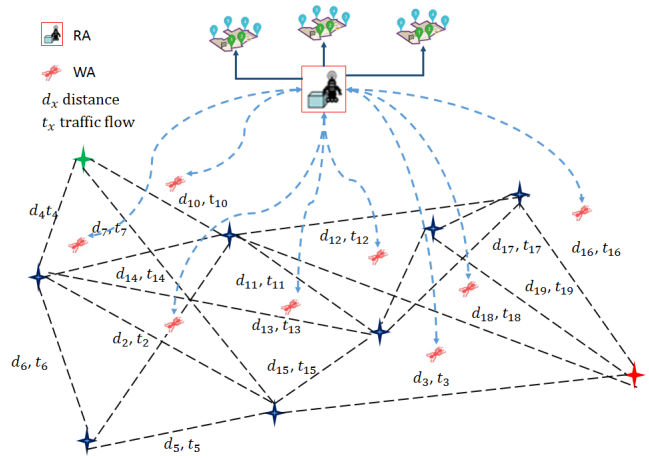

Fig. 6. Process of finding the best parking slot using the artificial ant agent

through the cloud platform system. The application is made based on Ionic framework that combines Apache Cordova and AngularJS framework. This hybrid application can run on both mobile systems (Android/iOS). The application is connected to our back-office system deployed on the dedicated server, the process of communication is based on HTTP communication using JSON format to transfer data between these two entities. The driver can ask for finding the nearest parking place which contains some available slots and follows the best path proposed by the remote application to arrive on the minimum time and avoid traffic jams.

VIII. SIMULATION AND RESULTS

In this section, we present the main functions of our distributed solution which is divided into two components: front-end / back-end layer, the first one is used by the driver to fetch and locate the set of nearest parking stations, this side is represented by a web application developed based on the HTML 5 and AngularJS framework and to deploy this application we are based on Apache Tomcat container as a virtual server characterized by the following properties:

- Processor information: Intel(R) Xeon(R) CPU E5-1650 v2 @ 3.50GHz, 4 cores

- Kernel and CPU: Linux 4.13.8-1.el7.elrepo.x86_64 on x86_64

- Operating system: CentOS Linux 7.4.1708

- Apache Webserver: 2.4

- Java Version: 1.8

- Google Maps Javascript v3.1

The main package (Fig. 7) block of our proposed solution is the hardware package which contains (Raspberry PI, Arduino, IR sensors), in this package, our sensors are configured to sense their surroundings and notify the Arduino module with the real-time state of parking slots. The number of available parking slots are transmitted to the Arduino module. Our sensors are configured and displayed on each slot inside the parking building to keep an open eye on the state of the parking places.

Fig. 5. Flow diagram to find a parking place for the driver

D. Vehicle Unit

This component is a hybrid application (Mobile/Web application) dedicated to being used as an interface of communication between the driver and the parking routing system
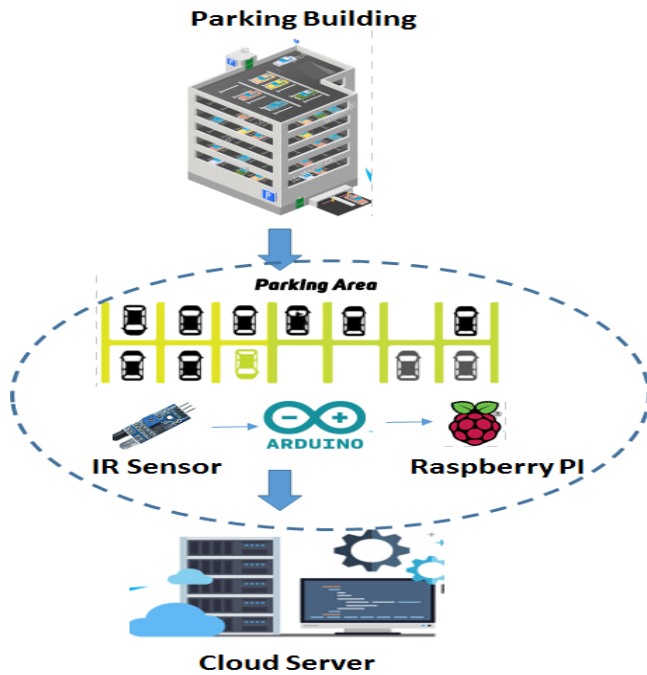
Fig. 7. Block diagram of the proposed smart parking system

To control the number of cars inside our parking building, we use another sensor placed at each entry of the building in the case when the parking has two entries. Each sensor will control the number of vehicles according to the triggering event when a vehicle passes through the detecting area of the sensor. To get the available parking slot, we placed at each slot an IR sensor, this one will be triggered when a car is inside his detecting area. After a car is detected by this sensor, it will send a notification push to the Arduino to update the state of this slot with two flags: occupied or available. On the other hand, we have the back-end side which is dedicated to compute and prepare all the data for the final user of this application, which is developed using the JADE (Java Agent Development Environment) that allow us to create a set of distributed agents, each one of them is able to work in parallel with other agents and has his private behaviour that presents the functions that will be done by this agent.

When the server handles the request of the driver, it will be able to create a circle around the GPS coordinates of the driver, then load all the existing parking around this circle, to fetch those parking stations we use our local database that contains the set of parking stations also we will show only parking with available slots, in our case, each parking station is viewed as a point-of-interest defined by the following attributes (latitude, longitude, available parking slots).

When the set of candidates parking stations are identified by our main agent, we can prepare our sub-graphs that contain the set of intersection between the actual location of the driver and the set of available parking stations. In Fig. 8, we can see the sub-road network created by the PFA, to put the state of traffic on this sub-area we are based on two remote services such as the RSU inside this area and the Google Traffic services. The data collected from the two services will be

added to our road network, this data represents the cost of traffic at each route on these sub-graphs. At the end of this step, we can see that our graph is fully prepared by the set of intersections that can be visited by the drivers which are represented by the parking places and the actual location of the driver. All routes on this sub-road network are characterized by two attributes (the distance in kilometres and the number of non-moving vehicles). In the next step, the PFA will be able to run the distributed solution which allows us to create a set of artificial ants to find the best paths between the actual location of the driver and each selected parking station on this area. At the end, the web application will show for the driver the set of routes toward each parking and give him the total length of the trip and the approximate travelling time to reach the parking according to the state of traffic on this area. So here we allow the driver to choose one of those trips.
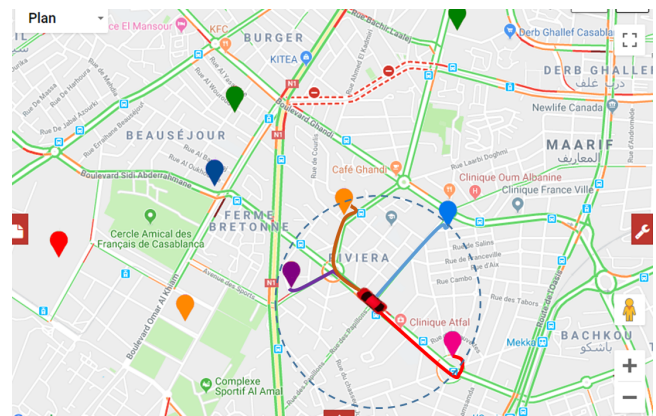

Fig. 8. Best path planning for a driver using the smart parking system strategy

Several scenarios have been made to see if our distributed strategy can be adopted in real life to help the driver into their daily needs. The first simulation has shown positive feedback and good results, but to see the efficiency of our method we should compare those results with other methods based on the same parameters to see if our strategy can give us useful feedback in terms of three constraints such as the length of the trip and the travelling time consumed on each trip and finally the time consumed by the prposed method to give as this solution (the best path between the driver location and the nearest parking station). As we can see that in this paper we are based on the techniques of swarm intelligent to build this solution, so we choose to compare our proposed solution with a combinatory method named as the Particle Swarm Optimization (PSO) [7] method to evaluate the results proposed by the two methods. The PSO was developed for continuous multi-objectives problems which try to find optimal solutions based on the collective behaviour of all those swarms methods [12], [28]. The PSO was proposed by Eberhart and Kennedy and was inspired by the behaviour of social bird flocking in nature. In the PSO algorithm, we are based on the mechanism of moving the better particles solutions locations in the solution space. The movement's system is controlled by the set of individuals and the best positions candidates. The PSO can propose the best solutions in a very fast convergence process due to a sample of two items such as memorizing individual best experiences (P_{best}) and information sharing of

the best global experiences (G_{best}). Also, the best-proposed solution G_{best} will be equal to the proposed solution of the best particle P_{best} in this solution space. Recently, this algorithm has been extended by adding the criteria of inertia weight. The application of this algorithm has covered multiple domains especially the vehicle routing problem which aims to find the set of alternative routes for a given set of vehicles by generating a kind of path planning. In this algorithm we are based on a set of sequences iterations, each one of them has two steps such as clustering the set of heuristic input data in our case the set of parking stations by applying the district PSO algorithm and secondly we will be able to call the Simulated Annealing (SA) algorithm to help as in the process of discovering the space solutions. But, based on this mechanism we will spend much time in case of solving large scale of problems, so many improvements have been made on this algorithm to upgrade them to give as effective solutions for combinatorial problems. This improvement has been done by injecting another swarm intelligent methods such as ACO, Genetic Algorithm (GA), Tabu Search algorithm (TS), each one of those algorithms is used in a specific step on the PSO algorithm iterations.

TABLE I. SIMULATION PARAMETERS FOR THE DISTRIBUTED PARKING SYSTEM (DPS)

Parameter	Value
Development environment	Eclipse Luna & JADE
Number of request	$[10 - 200]$
Road Topology	$[5 - 60]$
Number of artificial agents	$[10, 30, 60]$
Rate of evaporation ρ	0.1
Parameter Q	1
Parameter β	0.1
Parameter α	0.2

To evaluate the performance of our proposed strategy, we are based on twelves scenarios (bins) (each scenario is viewed as a graph made from a set of parking stations and routes), for each one of those bins, we evaluate our proposed strategy compared with the PSO algorithm to see the different result provided for each bin, the number of parking stations into each bin is limited because our strategy is applied in the city so the number of parking at a specific area will be limited. The result provided by each simulation describes the total distance of the trip between the current location of the driver toward each nearest parking station inside his area, the total time consumed to find the best trip for each map. So, we will find the best path that exists between the current position of the driver toward each selected parking station. In our case, we assume that a set of initial parameters are predefined by the configuration of the system such as the speed limit which is equal to 45 Km per hour because our system is applied in the city which means that the speed of the vehicle is limited. More parameters are used in our implementation to configure the AS algorithm which is based on a set of local parameters used for controlling the convergence of our program. In Table 1, we can see the rest of the parameters used for our simulation.

Firstly, we compare the performance of both solutions in terms of the total distance of each selected parking slot in the sub-area of the driver. The performance result proves that using our distributed strategy approach, a vehicle can reduce the

length of his trip only by following the shortest path proposed by this distributed method (Fig. 9) than the PSO approaches. Each scenario defines a road network combined with a set of parking stations and routes. The distance retrieved from the two approaches represents the total distance of the trip between the start and the destination of the driver. Due to the fact that the distributed approach uses more than one constraint to select the optimal route (such as the traffic data on the route and the linear distance) to propose the best paths and help the driver to reduce the length of his trip to achieve the nearest parking station, it performs better than the PSO algorithm. Simulation feedback, show that the proposed strategy works as well as the PSO algorithm for proposing the optimal path with the minimum travelling distance.

In the next level, we can see that the distributed system aim to reduce the travelling time consumed by the driver based on the AS algorithm by selecting only route with less traffic jam, and based on this mechanism we can reduce the total travelling time consumed by the vehicle to finish his trip. In our case we are based on a set of intelligent agents that implement a probabilistic rule which combine two parameters such as the distance and the traffic data, so, even if one route can be short in terms of distance we should evaluate the traffic jams on this route. We can say that based on this distributed solution we will search for the optimal one in terms of those two constraints (distance, traffic jam). As we can conclude based on the output of the system (Fig. 10), our distributed method reduce the rate of traveling time for all the given scenarios (for each size of bins) compared with the result provided by the PSO solution, because in this last one, we are based only on the linear distance between all the parking stations on each population and we not take into consideration the travelling time required to cross those nodes. So based on those elements, the distributed solution gives us the set of best paths that can be chosen by the driver for each nearest parking station inside the area of this driver.

Finally, we compare the required time consumed to compute and find the optimal solution for the given request by computing the time from handling the request toward the preparation of the best path that will be taken by the driver. The output data, show that our distributed strategy takes advantage of the parallel system to search and prepare the best solutions in a minimum time, also the strategy is based on a set of agents which they work in parallel to reduce the time consumed for arriving at the optimal solution, on the other hand, in the basic strategy (PSO Algorithm) we are based on a sequential process, which consumes more memory and CPU time. Compared with the PSO algorithm (Fig. 11), we can see that our proposed solution is better in terms of the quality of the proposed solution, also our system consumes less time to compute the best solution because of the parallel mechanism in our strategy.

As we can conclude and based on the feedback received from the results data we can see that the introduced method aims to solve the problem based on three levels :

- Searching for finding the best solutions based on the initial constraints (in our case we need to take into consideration the length of the trip and the traffic flow on this route).

- Reducing the time consumed to reach the selected parking stations which are related directly to the state of the traffic on the routes.

- Minimizing the time consumed to find those solutions because in our case we are using a distributed platform based on multiple agents, those agents are running in parallel to find and compute those best solutions, so the parallel behaviour is very important in our case to win mutch time when the system is started.

So the power of this method is represented by the distributed and the parallel process of communication between agents to compute and find the best solutions for this problem, also, in this recent year's several techniques have been made to improve the process of communication between entities which aim to reduce the time consumed for sharing useful data between them. Here, we are using this distributed behaviour to build a new concept of IoT combined with the ACO which will help us to define a new framework that implements the ACO on a distributed platform based on a set of intelligent agents.
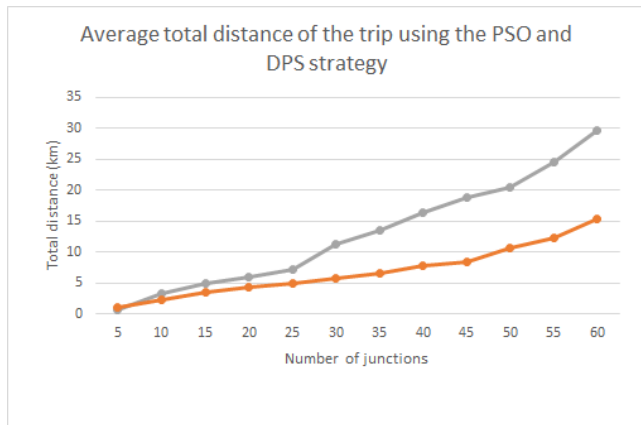
Fig. 9. Average total distance of the trip using the PSO and DPS strategy
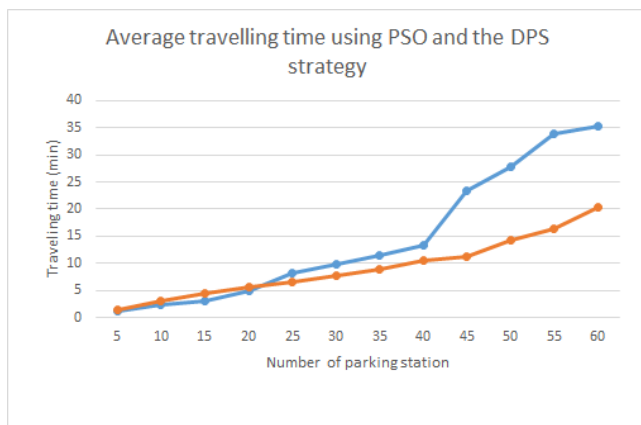
Fig. 10. Average travelling time between PSO and DPS

IX. CONCLUSION

The concept of building Smart Cities have always been a future vision for humanity and the growth of Cloud technologies and the Internet of Things have given rise to discover
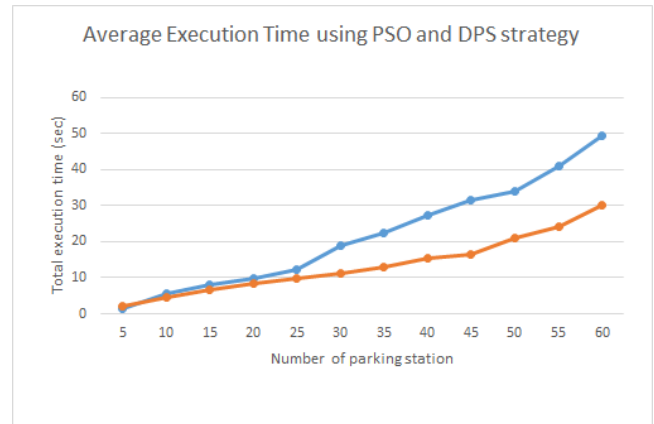
Fig. 11. Comparaison of average execution time to find the optimal solution between PSO and DPS

new possibilities to improve the services provided by our cities and transfer them to a smart cities. In this paper, we tried to create a new strategy of the parking routing system based on the internet of things combined with a distributed swarm intelligent method to improve the quality of the proposed solutions. At this position, the presented solution aims to help drivers to find the nearest parking places by taken into consideration a set of constraints such as the total travelling time to achieve this parking and the state of traffic road to reach this parking station, so based on these constraints, our solution implements a distributed architecture which contains a set of distributed intelligent agents each one of them has a predefined behaviour. The role of this system is to find the best shortest paths between the location of the driver and the available parking stations. As shown in the result section, we can said that our system will be a useful solution for driver to find the nearest parking slot in a given area and give them the best path planning to reach this parking based on the technique of the ant colony optimization which is dedicated for solving problem attached to the vehicle routing system.

REFERENCES

[1] Ameya Naik, Sheetal Vatari, Manjiri Gogate; Real time traffic management using Internet of Things,Tanvi Tushar Thakur ; International Conference on Communication and Signal Processing (ICCSP),2016

[2] Ankit Dubey, Mayuri Lakhani, Shivansh Dave, Jignesh J. Patoliya; Internet of Things based adaptive traffic management system as a part of Intelligent Transportation System (ITS); 2017 International Conference on Soft Computing and its Engineering Applications (icSoftComp)

[3] Aws Naser Jaber et al; A study in data security in cloud computing ;2014 International Conference on Computer, Communications, and Control Technology

[4] B M Mahendra, Savita Sonoli, Nagaraj Bhat, Raju, T Raghu; IoT based sensor enabled smart car parking for advanced driver assistance system; 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)

[5] Guy Krasner , Eyal Katz; Automatic parking identification and vehicle guidance with road awareness; 2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE)

[6] Hong Zhan, Zhigang Wen , Yuxin Wu , Junwei Zou , Shan Li; A GPS navigation system based on the internet of Things platform;IEEE 2nd International Conference on Software Engineering and Service Science,2011

[7] Hong Guo , Dandan Han , Hongguo Zhang; Using PSO to improve ant colony optimization algorithm;International Conference on Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things 2014

[8] Ilhan Aydin, Mehmet Karakose, Ebru Karakose; A navigation and reservation based smart parking platform using genetic optimization for smart cities; 2017 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG)

[9] Jerry Kponyo, Yujun Kung, Enzhan Zhang; Dynamic Travel Path Optimization System Using Ant Colony Optimization;2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation

[10] Jerry John Kponyo, Yujun Kuang, Zejiao Li; Real time status collection and dynamic vehicular traffic control using Ant Colony Optimization;2012 International Conference on Computational Problem-Solving (ICCP)

[11] Kai Lin , Chensi Li, Giancarlo Fortino , Joel J. P. C. Rodrigues; Vehicle Route Selection Based on Game Evolution in Social Internet of Vehicles;IEEE Internet of Things Journal (Volume: 5 , Issue: 4 , Aug. 2018)

[12] Kui-Ting Chen, Yijun Dai, Ke Fan, Takaaki Baba; A particle swarm optimization with adaptive multi-swarm strategy for capacitated vehicle routing problem; 2015 1st International Conference on Industrial Networks and Intelligent Systems (INISCom)

[13] Muhamad Asvial , M. Faridz Gita Pandoyo , Ajib Setyo Arifin; Internet of Things Solution for Motorcycle Riders to Overcome Traffic Jam in Jakarta Using EBkSP;2018 International Conference on Information and Communication Technology Convergence (ICTC)

[14] Meijuan Kou, Yanlin Zhao, Hanyu Cai, Xiumei Fan; Study of a Routing Algorithm of Internet of Vehicles Based on Selfishness;2018 IEEE International Conference on Smart Internet of Things (SmartIoT)

[15] M.Elgarej, K.Mansouri, M.Youssfi, N.Benmoussa, H.elfazazi.(2017) "Distributed Swarm Optimization Modeling for Waste Collection Vehicle Routing Problem", International Journal of Advanced Computer Science and Applications(ijacsa)

[16] Priti Chakurkar, Sajeeda Shikalgar, Debajyoti Mukhopadhyay; An Internet of Things (IOT) based monitoring system for efficient milk distribution;2017 International Conference on Advances in Computing, Communication and Control (ICAC3)

[17] Pranav Chippalkatti, Ganesh Kadam, Vrushali Ichake; I-SPARK: IoT Based Smart Parking System; 2018 International Conference On Advances in Communication and Computing Technology (ICACCT)

[18] Prabhu Ramaswamy; IoT smart parking system for reducing green house gas emission; 2016 International Conference on Recent Trends in Information Technology (ICRTIT)

[19] Sangita S. Chaudhari, Varsha Y. Bhole; Solid Waste Collection as a Service using IoT-Solution for Smart Cities;2018 International Conference on Smart City and Emerging Technology (ICSCET)

[20] Sokwoo Rhee; Catalyzing the Internet of Things and smart cities: Global City Teams Challenge; 2016 1st International Workshop on Science of Smart City Operations and Platforms Engineering (SCOPE) in partnership with Global City Teams Challenge (GCTC) (SCOPE - GCTC)

[21] Tanmay Chakraborty , Soumya Kanti Datta; Application of swarm intelligence in Internet of Things;2017 IEEE International Symposium on Consumer Electronics (ISCE)

[22] Vasile Gheorghiţă Găitan, Corneliu Octavian Turcu; An internet of things-based distributed intelligent system with self-optimization for controlling traffic-light intersections,Cristina Elena Turcu;International Conference on Applied and Theoretical Electricity (ICATE),2012

[23] Xiaoying Tang , Suzhi Bi , Ying-Jun Angela Zhang; Distributed Routing and Charging Scheduling Optimization for Internet of Electric Vehicles;IEEE Internet of Things Journal (Volume: 6 , Issue: 1 , Feb. 2019)

[24] Xiaobo Zhang, Liangjie Yu, Yong Wang, Guangqing Xue, Yanbo Xu; Intelligent travel and parking guidance system based on Internet of vehicle; 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)

[25] Xing Xiong, Byung-Jae Choi; Design of Genetic Algorithm-based Parking System for an Autonomous Vehicle; International Journal of Fuzzy Logic and Intelligent Systems, vol. 9, no. 4, December 2009 pp. 275-280

[26] Yang Haoxiong , Hu Yang; Congested traffic based on ant colony algorithm for shortest path algorithm; 2015 International Conference on Logistics, Informatics and Service Sciences (LISS)

[27] Yonca Erdem Demirtaş, Erhan Özdemir, Umut Demirtaş; A particle swarm optimization for the dynamic vehicle routing problem;2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)

[28] Zhuangkuo Li, Yannan Ma; Particle Swarm Optimization Based on the Average Optimal Information for Vehicle Routing Problem;2013 Sixth International Symposium on Computational Intelligence and Design