IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 11, 2019, accepted July 17, 2019, date of publication July 23, 2019, date of current version August 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2930564

AVPark: Reservation and Cost Optimization-Based Cyber-Physical System for Long-Range Autonomous Valet Parking (L-AVP)

MUHAMMAD KHALID[1], YUE CAO[2], NAUMAN ASLAM[1], MOHSIN RAZA[3], ALUN MOON[1], AND HUAN ZHOU[4]

[1]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K.
[2]School of Computing and Communications, Lancaster University, Lancashire LA1 4WA, U.K.
[3]Department of Design Engineering and Mathematics, Middlesex University, London NW4 4BT, U.K.
[4]College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China

Corresponding author: Huan Zhou (zhouhuan117@gmail.com)

ABSTRACT The autonomous vehicle (AV) is an emerging product of intelligent transportation system. This paper proposes a new parking cost optimization scheme for long-range autonomous valet parking (L-AVP), namely AVPark. The L-AVP selects a drop-off point (as the temporary reference point for people to fetch the AV for traveling purpose) for AV. The user leaves AV at drop-off spot and the AV finds out the most optimal car parks (CPs) itself. The AVPark provides an AV with the most optimal car park considering the parking price, fuel consumption, and distance to a vacant parking space. AVPark aims to minimize the walking distance for drivers, and also the round-trip duration for AV from the drop-off point to car park through a combination of weighted values and heuristic approach. By facilitating the drop-off point that is newly brought into the emerging scenario, an optimization scheme is proposed to minimize the total cost for fuel consumption and traveling time using the weighted value analysis. The results show that AVPark optimized the total trip duration, walking distance, and cost.

INDEX TERMS Autonomous parking, optimization, autonomous driving, reservation.

I. INTRODUCTION

Transportation system has always remained an important aspect of human life. Human mobility is largely dependent on the transportation and infrastructure. According to transportation statistics of Great Britain 2017 [1], 78% of distance is covered by own transport while remaining 22% is covered by other means of public transport. The other means of public transport includes bus, train and cabs. Similarly, in the USA, Europe and China, distance travelled by private and public transportation mean is illustrated in Fig. 1. Recent statistics show that there is an increase in travelling through private mode of transportation.

Usually in urban areas, there is only a limited number of parking spaces available. On average, vehicles searching for parking space contributes to 30% of traffic on roads. In the UK, it takes over 6 minutes to find a parking lot, in a report by JustPark [2]. In a global parking survey by IBM [3], 20 minutes are spent in searching for an appropriate CPs [4]–[6]. Parking a car in a pre-defined and limited space itself is a difficult task for most of the early stage drivers. As per the recent report of Insurance Institute for Highway Safety (IIHS), 20% of all car accident occurs in commercial parking lots. The rate of auto-mobiles has been doubled over the past few decades and it is quite challenging to accommodate increasing number of vehicles in existing infrastructure. Besides limited space for parking, skills required to manoeuvre vehicle in tight spaces, high cost and circular driving are various reasons that need to be addressed using the state of the art technologies.

On one hand, a personal vehicle provides much comfort. On the other hand, one may face multiple challenges while driving their own vehicle. One of the most challenging tasks is to find a cheap and suitable parking space in a timely manner. Smart Parking (SP) benefits to user, by providing suggestion for CPs. Apart from finding parking slot, parking vehicle into exact dimensions specified for parking is considered as the most difficult part of all driving operations. These days,

The associate editor coordinating the review of this manuscript and approving it for publication was Liang Hu.

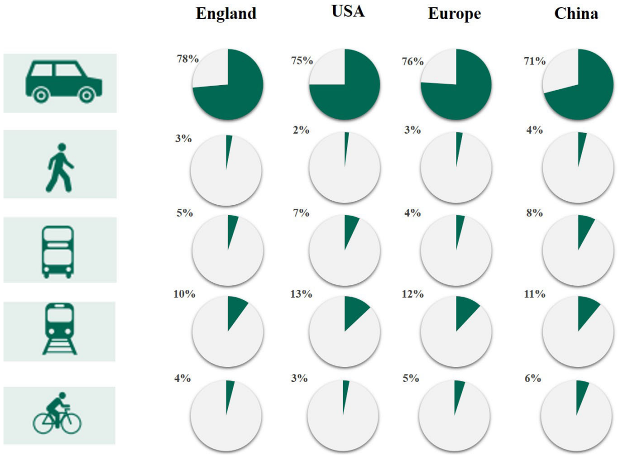

FIGURE 1. Statistics of 2017-2018 [1], [2].

finding CPs in urban areas, congested zones, business areas, and tourist spots is a major concern due to the increase in number of vehicles.

Recent inventions in autonomous control systems, sensing and vision systems have addressed many of the major challenges with respect to safety on wheels, efficiency, fuel consumption, traffic congestion and environmental pollution [7]. The AV provides the capacity of performing multiple operations at the same time autonomously [8]. It includes several operations such as parking vehicle in the garage without a driver, searching for a vacant parking slot in CPs and getting parked.

In Autonomous Valet Parking (AVP), the driver leaves an AV at designated locations. The AV is capable of driving in fully autonomous mode, moves towards selected CPs and gets parked. The AVP is delivering astonishing services with the help of modern automation technologies. It improves the overall user experience and provides safety as well [9].

One type of AVP called Short range Autonomous Valet Parking (S-AVP) has been made possible through the advancement of vision system and autonomous car-maneuvering techniques [9]. The S-AVP has addressed various problems that happen in parking lots, like finding empty parking lot inside CPs and parking AV in specified dimensions. The S-AVP has performed efficiently in training scenarios, where AV training is performed by the driver at least once. The most advanced system in S-AVP can search for vacant parking lots in multi-story buildings. Through advanced machine learning techniques, S-AVP can park an AV in full autonomous mode [10], [11].

This paper will address key challenges related to Long-range Autonomous Valet Parking (L-AVP), which is an extension of the existing S-AVP. In L-AVP, the user drives the AV at a certain point in city center called Drop-off/Pick-up (D/P) point and walks towards the desired location, which could be a Leisure Point (LP), or a Work Place (WP). In the next step, the AV moves autonomously towards CPs. Similarly, for the inbound trip from CPs, the AV is picked-up by the driver at pick-up point. The L-AVP provides driver with more convenience, by allowing them to drop their AVs near LP. The proposed scheme also saves significant amount of time because, L-AVP allows drivers to be dropped off at nearest point to their desired location. L-AVP also facilitates smart operations executed through AVs by providing the most suitable path planning, cost effective selection of CPs and suggestions for pick-up/drop-off spots selection.

In literature, pricing schemes for smart parking has been proposed and analyzed [12]. The L-AVP delivers by providing the nearest drop-off point near to their work place/leisure point. The AV is facilitated at the nearest drop-off point, and provides users the convenience in terms of time and monetary value. Our contribution focuses on providing efficient pricing and reservation scheme for L-AVP. The main contributions of this paper are as follows;

- **Optimal Parking Cost** The work in [12] is more towards resource[1] allocation. To provide user with more convenience and economic solution for parking, this work aims to minimize parking cost[2] by comparing suggested cost with already set minimum value.
- **Drop-off Spot & Car Park Recommendation** The work in [13] used only distance parameter to analyze the best CPs. This work recommends the best D/P by calculating minimum distance from D/P to WP. Also, the distance between current position of AV and suggested D/P is minimized. A cost function is used to recommend most appropriate CPs.

II. BACKGROUND

A. TRADITIONAL PARKING

A couple of years ago, pre-parking information about parking places, their prices, and location were used to be almost null [14]. There was no prior information about where to park, how much it will cost and, how far it is. In traditional parking, a driver has to check each parking lot in search of a free CP slot. On one hand, it costs them in terms of time, fuel consumption and the hassle of moving the vehicle in CPs multiple times. While on the other hand, it tends to produce greater environmental pollution [15], [16].

B. SMART PARKING (SP)

A general Smart Parking (SP) model consists of following five elements [17]–[19]. User interface module connects the user with parking servers and manager module. The user can perform multiple operations through this interface. The communication module ensures availability and reliability of overall communication of the system. The communication module performs multiple operations like encryption, controlling errors, exchanging data and information reliably. The parking module looks after all operations inside the parking area. It analyses the vehicles and parking slots. Space controller unit consists of a combination of sensors and a

[1] Resource refers to vacant car park slot

[2] Parking cost includes the fuel consumed and price paid for car park

controller. These sensors detect a vehicle in the parking lot and inform the controller about parking slot status. The manager module is responsible to take care of parking servers and information about registered users.

The i-Parker parking scheme is mainly based on intelligently allocating resources, defining prices and reserve parking lots when necessary [12]. The main contribution of i-Parker is the confirmed reservations with the lowest parking price and searching time. The i-parker combines the concept of real time reservation and share time reservation. Real time reservation and share time reservation allow users to select parking spot any time earlier or on the spot. Real time reservation continuously allocates vehicle with the best parking slot. While in share time reservation, user selects a specific parking and time slot as per their convenience. This scheme allows users to select multiple parking slots at the same time and the system will recommend the best parking space as per the current circumstances. The architecture of this system has been categorized into central request center, parking manager and smart allocation systems. The purpose of central request center is to receive parking request and then put it forward to appropriate smart allocation systems for allocation of parking space. The parking manager acts as an interface between smart allocation systems and parking authorities. smart allocation system consists of pricing engine, sensors, data center, smart allocation center and virtual message sign. The queuing model is divided into dynamic and static parking. The i-Parker uses mixed integer linear programming to minimize the monetary cost for the users. The mixed integer linear programming also focuses on minimum resource utilization with higher revenue generation.

The SP system assigns and reserves an optimal parking lot depending on user's cost function [20]. The components of this system include parking geographic information system, driver requests processing center, parking resource management center and smart parking allocation center. The parking resource management center updates real-time parking information and delivers it to end users via the internet or virtual message sign. Similarly, driver request processing center is responsible for collecting users request for parking, and keeps an eye on the recent allocation of resources to users. This system takes the current road condition and parking space information into account, and suggests an optimal parking solution. It reserves an optimal parking space for the user. In earlier parking mechanisms, the suggested space was been occupied by some other vehicle when the actual vehicle reaches. An objective function is used to compute an optimal parking space as per user needs. The user's objective function, depending on the distance to CPs and parking cost. The proposed algorithm solves mixed integer linear programming problem at every decision point. The mixed integer linear programming proposes an optimal parking slot on the basis of data being provided by the user.

Campus parking system focuses on the efficient use of the existing parking slots on sharing basis by dividing users into day and night shifts [21]. This scheme is provide a general framework which could be deployed for various scenarios. The number of supposed parking slots on campus are k while l are parking slots on private parking are in surrounding area. The proposed scheme has supposed that k parking slots can be utilized by university staff and students once l parking slots are filled. It is considered the k parking slots are utilized at night time and these people go to offices in day time. In this system, it has been considered that a certain contract exists between university administration and private parking owners.

C. AUTONOMOUS VALET PARKING (AVP)

The Fig. 2 gives a bird-eye view of the properties of both SP and AVP. There are a few features that makes AVP a novel and efficient technique for parking than SP. In early days, AVP was used to provide a limited parking assistance. Automatic parking can be used if the driver remains inside the AV. This process is not fully autonomous as driver can intervene during the process. In this process, the whole parking activity is fully supervised by the driver, referred to as Level 1b in Fig. 3. The driver drives AV to a vacant parking slot and set AV position at a certain distance from the obstacle and other AVs. Once AV is in heading position towards the parking slot, it automatically detects the slot and get parked. This mechanism is mostly useful for less experienced drivers and it has minimum chances of hitting an obstacle or another AV.

In the following years, AVP developed wireless operations. It enables the driver to stay out of the car, perform and monitor parking process through their specified handset or smart phone, which is referred as ''level 2''. In the later stage which is shown as ''level 3'', 3D mapping and sensing technologies are used. This is a more advanced level of AVP, where AV travels to parking lot from a specific spot. Usually, in this technique, an AV is trained at least once with driver inside AV [22].

In the state-of-the-art AV systems, path generation [23] and precise detection [24] techniques have extended the applications of AVP. In this scenario referred as ''Level 4a'', a driver leaves AV at CPs entrance and navigates AV towards a vacant slot [25]. The disadvantage of this system is that driver must approach CPs and drop AV there, however it saves time to find parking slot. User leaves AV at CPs, while the AV autonomously searches for an available parking slot.

Short-range Autonomous Valet Parking The recent development in machine vision system and autonomous car-manoeuvring techniques has made it possible for S-AVP to become a mature technique. The S-AVP has addressed many parking issues where space for parking operations is found to be limited. For S-AVP, AV must be trained at least once to adopt the new routes and analyze the obstacles [26]. Recent advancement in machine learning techniques can make it capable of achieving optimal output for S-AVP in complex parking areas [27], [28].

In initial steps, AV is parked by the driver to train AV and familiarizes it with the new area and surroundings.

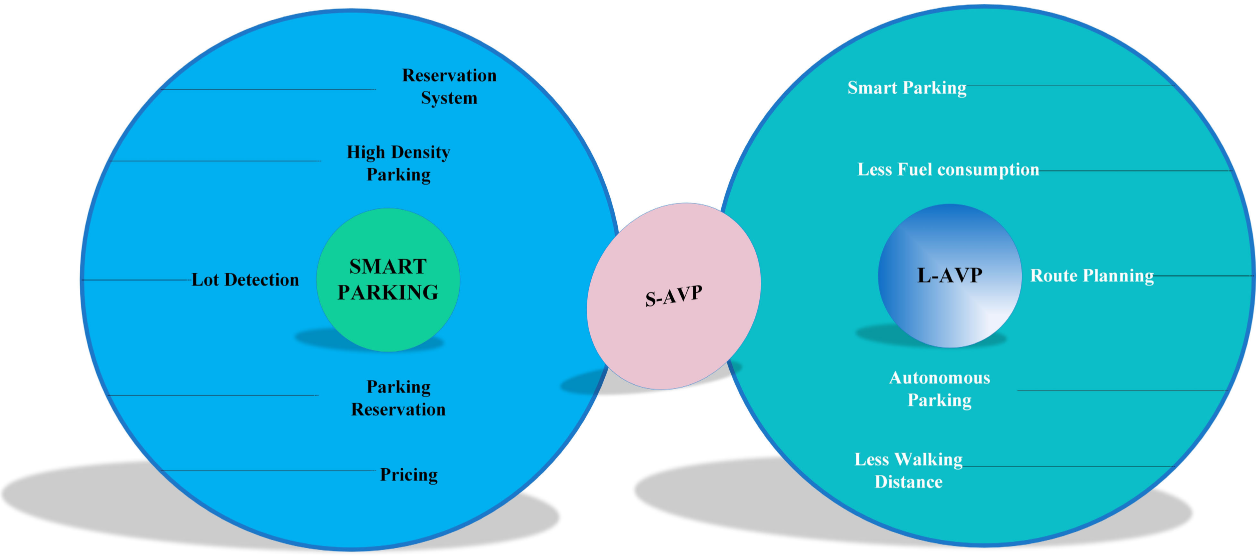

FIGURE 2. From smart parking to autonomous valet parking.

FIGURE 3. AVP at low autonomous level.

This operation is performed at least once by the driver to supervise AV properly. In the following step, AV is parked in complete autonomous mode. The recent advancement enables an AV to scan for objects coming its way and perform accordingly. Nowadays, AV is equipped with multiple sensors performing different kinds of functionalities. Due to these advancements, S-AVP is as mature. S-AVP can search for a vacant parking slot, avoid objects coming its way and the most important is to take care of living objects while performing autonomous driving. In the next step, AV learns to park itself without a driver. Advancement in AVP system has made it capable of scanning for available CPs, slot by slot and floor by floor in case of multi-story CPs. AVP has made AV capable of parking in fully autonomous mode [29]. The work in [8] provides valuable recommendation on parking slots status, that results in saving time. In [30], studies about coordination of AV to available parking facilities in Vehicle-to-Grid (V2G) environment for electric vehicles.

III. SYSTEM FRAMEWORK

In L-AVP, multiple D/P are deployed around the city, while CPs are deployed in remote area or border line of the city. When an AV (in driving mode) is serving users, it searches for the nearest D/P. For example, in the out bound trip, the user leaves AV at a drop-off spot, and walks towards WP. Then the AV travels towards CPs and finds a suitable parking slot. Similarly, for inbound trip, the drop-off spot would act as a pick-up spot. The AV is able to pick-up user and delivers towards inbound trip destination [31].

A. AUTONOMOUS VALET PARKING MODEL

The AVP design serves as core model for the autonomous parking process. All the notations used in Section III are defined in Table 1. The model includes various modules related to parking process as following and as presented in Fig. 4;

1) AUTONOMOUS VEHICLE

AV is supposed to be the end user of AVP model. An AV can directly communicate with Resource Request Centre (RRC) and the Vehicle-to-Everything (V2X) infrastructure. The V2X acts as primary source real-time information for AV. The RRC is liable to receive any parking request from AV and process them. The AV is able to receive city wide CPs information through V2X and similarly, AV can request for parking resources to RRC. The AV uses RRC to request for any available resources. It receives real-time information about surrounding region from V2X. This information includes congestion rate, routes and CPs information in the region being covered by that specific V2X.

2) RESOURCE REQUEST CENTRE

The responsibility of the Resource Request Centre (RRC) is to receive request from AV and forward it to the Reservation Centre: 1) Pre-arrival Reservation; 2) Real-time Reservation.

TABLE 1. Main notations.

α	Vehicle acceleration in m/s^2
v	Vehicle speed assuming no headwind, in m/s
g	Acceleration due to gravity $9.8m/s^2$
G	Road grade
C_R	Rolling resistance 0.009
ρ	Air density 1.2 kg/m^3
C_D	Aerodynamic drag coefficient 0.3
A_F	Frontal area $2m^2$
ζ	Mass factor accounting for the rotational masses 0.1
m	Vehicle mass in metric tonnes (1305 kg for AV, plus 80 for driver)
P	Fuel consumption
P_{avg}	Average fuel consumption
ξ	Number of hour(s)
y	Fuel consumption weight coefficient
z	Parking cost weight coefficient
$J_{lavp-out}$	Total L-AVP journey cost (outward)
$J_{bck-out}$	Total benchmark journey cost (outward)
$D_{v,cps}$	Distance from AV current location car park
$D_{cps,v}$	Distance from car park to destination
S_n	n number of total D/P spots
sl_n	n number of slots in each CPs
P_{min}	Possible minimum fuel consumption in reaching CPs
C_ξ	Parking fee for number of ξ hours
ξ	Number of hours for which parking slot is reserved
Υ_{op}	Optimized parking cost
Υ_{avg}	Average parking cost
Υ_{max}	Maximum parking cost
$U_{cps,w}$	Travelling cost for user from CPs to WP
M	Upper limit for Drop-off spots
N	Upper limit for CPs slots
dp_M^{curr}	Currently available drop-off spot out of M spots
cps_N^{curr}	Currently available CPs out of N CPs
T_{lavp}^{out}	L-AVP outbound trip time
T_{bck}^{out}	L-AVP outbound trip time
$D_{v,d}$	Distance between vehicle and D/P spot
S_v	Speed of Vehicle
$D_{d,w}$	Distance between D/P spot to WP
S_h	Speed of human

3) RESERVATION CENTRE

- Pre-Arrival: In pre-arrival, a fixed slot is assigned to AV at a specified CPs, if available.
- Real-time: A real time reservation is made when an AV is about to reach a parking space. It is updated continuously by the server to provide AV with efficient cost and better experience.

4) COST CONTROL CENTRE

The cost control center deals with parking prices for different CPs. The cost control center change prices as per traffic situation and to make CPs for profitable.

- Cost Centre: This module sets the prices of parking slots in a region covered by an V2X. The parking price is set depending on the demand of parking slots. Usually, a smart approach is followed to benefit the operators of CPs as well as AV.
- Cost Optimizer: The cost optimizer tends to provide users with minimum possible parking fee. It calculates the average parking fee and aims to provide users with below average parking prices.

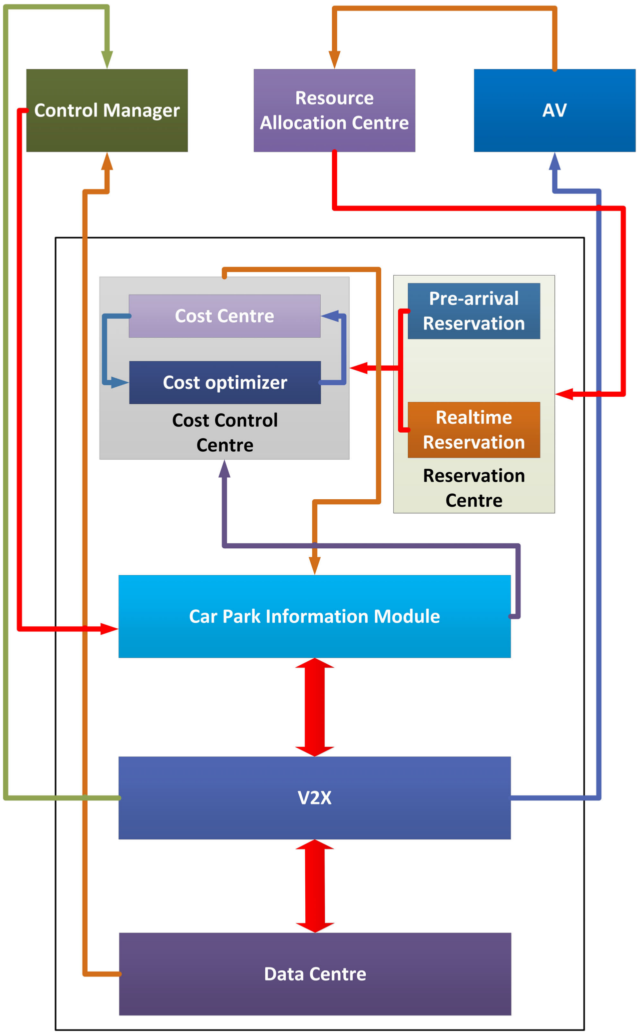

FIGURE 4. AVP model.

5) CAR PARK INFORMATION MODULE

This module is responsible to keep track of all the CPs. It includes CPs current vacant and occupied slots, reserved slots and their predicted status in the near future. The car park information module directly communicates with Cost center and V2X.

6) V2X

It acts as an intermediate unit between CPs module and data center. It also communicates and provides CPs information to AV directly. The V2X is required to cover a specific geographic area and report it to the data center. Similarly, the V2X system is connected to data center at the same time. Usually, all the computation is carried out on this end and results are forwarded to data center.

7) DATA CENTRE

This is the main module where all the relevant data about CPs, parking fee, routes, vehicles and traffic condition is held.

It can directly communicate with V2X and receive data frequently, in order to update other V2X accordingly.

8) CONTROL MANAGER

The control manager module is responsible to monitor overall system. It observes the parking fee and makes changes to provide CPs owner's with maximum output. The control manager offers lower parking fee to users, in order to enhance their parking experience.

B. LONG-RANGE AUTONOMOUS VALET PARKING

1) L-AVP COMMUNICATION

In this framework, all AVs can communicate with network entities introduced in section III-A through V2X infrastructure, as depicted in Fig. 5.

2) DROP-OFF SPOT SELECTION & RESERVATION

Initially, AV is in travelling mode. In the Algorithm 1, the process of searching and selecting a drop-off spot has been explained. All the available M spots are extracted in the initial stage. As explained between lines 4 to 10, the minimum distance as a summation of $D_{v,d}$ (from location of AV to D/P) and $D_{d,w}$ (from D/P to WP) is ranked, in case $M > 0$. Once reserved, the reserved spot is deducted through $M = M - 1$.

Algorithm 1 Reservation of D/P Spot

```
 1: Define MIN, $\aleph$
 2: if ($M > 0$) then
 3:     for ($i = 1; i \leq M; i++$) do
 4:         Calculate $\left(D_{v,d_{(i)}} + D_{d_{(i)},w}\right)$
 5:         if $\left(\left(D_{v,d_{(i)}} + D_{d_{(i)},w}\right) < \text{MIN}\right)$ then
 6:             $\text{MIN} = \left(D_{v,d_{(i)}} + D_{d_{(i)},w}\right)$
 7:             $\aleph = i$
 8:         end if
 9:     end for
10:     Select $D/P_{\aleph}$
11:     $M = M - 1$
12: end if
```

3) CAR PARK SELECTION & SLOT RESERVATION

The Algorithm 2 aims to address the challenge of CPs selection. It extracts the *CP.LIST* from the nearest V2X. Each AV is treated on First Come First Serve (FCFS) basis for CPs selection. Each CPs in the *CP.LIST* is checked for three conditions for available slots, where N is upper bound for number of slots in each CPs. Here, $(cps_N^{curr} = N)$ is considered as the best option, $(N > cps_N^{curr} \geq 1)$ as average and $(cps_N^{curr} = 1)$ means that only one slot is left in selected CPs. They are compared on average cost of P as fuel consumption and C as the parking cost in the CPs. The cost is calculated through Υ in line 12.

Algorithm 3 helps AV in reserving a slot in selected CPs defined by Algorithm 2. It checks for the available slot in the CPs, considering the total cost and distance.

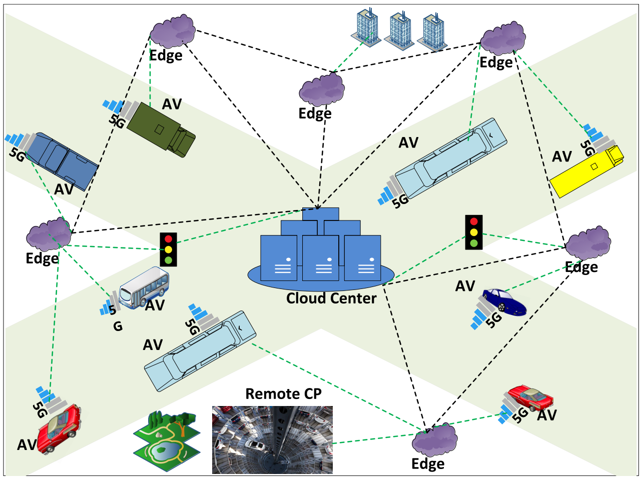

FIGURE 5. L-AVP communication framework.

Algorithm 2 Car Park Selection

```
 1: for Extract $CP.LIST$ do
 2:     for Serve every AV's on FCFS do
 3:         Get data about available slot on each CPs
 4:         for ($j = 1; j \leq N; j++$) do
 5:             if ($cps_N^{curr} = N$) then
 6:                 Select CPs with $C_{(j)} < 0.5$ & lowest $P_{(j)}$ value
 7:             else if ($N > cps_N^{curr} \geq 1$) then
 8:                 Select CPs with $C_{(j)} \& P_{(j)} \leq average$
 9:             else
10:                 Select last available CPs
11:             end if
12:             Calculate $\Upsilon = P_{(j)} + C_{\xi(j)}$
13:         end for
14:     end for
15: end for
```

The Algorithm 3 computes the minimum value of parking Υ_{op}. Once reserved, it is removed from available slot list $sl_n = sl_n - 1$. Here, sl_n is number of total slots in certain CPs. The status of CPs is updated through cps_N^{curr}.

C. PARKING COST OPTIMIZATION

Initially, vehicle is in driving mode. The moment driver intends to park a vehicle, a parking request is initiated and nearest drop-off spot is selected, if available. After selection of available drop-off spot, CPs is checked for vacant slots. A list of CPs having vacant slots is obtained and selection of CPs is made by the user.

The fuel consumed from drop-off zone to CPs can be calculated by P [32],

$$P = mv[\alpha(1 + \zeta) + gG + gC_R] + 0.5\rho C_D A_{Fv^3} \qquad (1)$$

Let total cost of parking be denoted by Υ. It can be achieved by calculating fuel consumption P and parking fee per hour. Here, C be the parking fee per hour while ξ as the number of hours vehicle is being parked. So the total parking

Algorithm 3 Slot Reservation in CPs

```
for Get All Available CPs do
    Get $CP.LIST$
    for Each CPs in $CP.LIST$ do
        if $(0 < cps_N^{curr} \leq N)$ then
            Reserve slot having the minimum cost $\Upsilon_{op}$
        else if $(1 \leq cps_N^{curr} \leq 1)$ then
            Calculate cost
            Reserve slot
        else
            Select last available slot
        end if
        $sl_n = sl_n - 1$
        slot reserved
        Update $cps_N^{curr}$
    end for
end for
```

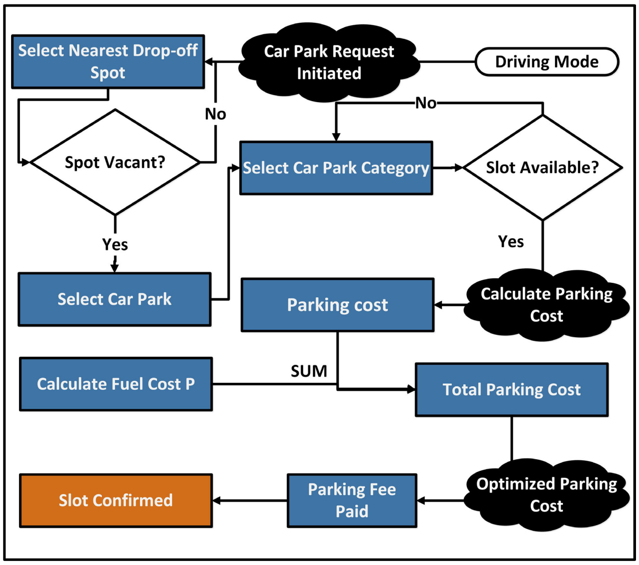

FIGURE 6. Cost optimization flow chart.

cost along the time is, $\sum_{\xi=1}^{\varrho} C_\xi$,

$$\Upsilon_{max} = P + \sum_{\xi=1}^{\varrho} C_\xi \tag{2}$$

Considering the total cost of parking, following elements are considerable;

$$P, C, S_v, S_h \tag{3}$$

This model uses a weighed value for fuel consumption and parking fee, keeping in view fuel and parking prices. For fuel consumption P, a weighted value of y is assigned while a weighted value of z is assigned to parking fee C and ξ is number of hours a slot is need, where $\xi = \{1, 2, 3...., 12\}$. Here, we denotes the maximum number by ϱ. The S_v is speed of AV and S_h is speed of human.

$$\Upsilon_{max} = y(P) + z\sum_{\xi=1}^{\varrho} C_\xi \tag{4}$$

$$\sum_{\xi=1}^{\varrho} \Upsilon_{max} = y(P) + z\sum_{\xi=1}^{\varrho} C_\xi \tag{5}$$

The slot can only be reserved for up to 12 hours. If a user wants a slot for more than 12 hours, a re-reservation process must be performed to reserve the same slot again.

$$\Upsilon_{avg} = P_{avg} + C_\xi \tag{6}$$

Here, the term avg can be defined as $(0.01 \leq avg \leq 0.5)$ Similarly, for up to 12 hours parking slot the equation can be given as follows;

$$\sum_{\xi=1}^{\varrho} \Upsilon_{avg} = \{P_{avg} + \sum_{\xi=1}^{\varrho} C_\xi\} \tag{7}$$

Υ_{max} is compared with the average parking cost for AV Υ_{avg}. Here, v is considered as number of AV's and $v = \{1, 2, 3 .... n\}$ Similarly, for n number of AV's, the equation can be given as;

$$\sum_{v=1}^{n} \Upsilon_{op} = \sum_{v=1}^{n} \left\{\frac{\Upsilon_{avg}}{\Upsilon_{max}}\right\} \quad where\ (0 < \Upsilon_{op} \leq 1) \tag{8}$$

The value of Υ_{op} will lie between 0.01 & 1.0. The value of Υ_{op} vary depending on traffic condition and usage of parking spaces. It also depends on how busy the city is. The more a city center is busy, the higher parking price would be. Although the cost of fuel will have a rare effect of overall cost. This model aims to serve user with the minimum value of Υ_{op}. The lower the value of Υ_{op} is, the minimum will be the parking price. The goal of algorithm 4 is to optimize the existing parking price.[3] This algorithm first extracts $CP.LIST$ and analyze the average cost for parking in specific area by $\Upsilon_{avg} = P_{avg} + C_\xi$. The P_{avg} is then compared with maximum and minimum possible cost. The parking slot detail each CPs is obtained through $sl.LIST$. For the optimal parking cost $\Upsilon_{op} = \left\{\frac{\Upsilon_{avg}}{\Upsilon_{max}}\right\}$, a normalized value is calculated. After calculating Υ_{op} the lowest value among all the obtained values is selected. The Υ_{op} can only be obtained when slots in selected CPs qualifies are greater than $\frac{n}{2}$ and less than or equal to sl_n, which are total available slots in each CPs. The average value of parking cost Υ_{avg} can only be obtained when $\frac{n}{2}$ or less slots in selected CPs are available.

D. AVPARK & BENCHMARK

A detailed study and analysis is provided to analyze the advantages of L-AVP. As inbound and outbound trips are considered as same, so outbound trips are considered for analysis. For benchmark, D/P spots are not considered. As user would

[3] Here, the price includes parking price for using a parking slot in CPs and the fuel cost for the distance covered by AV to reach that specific CPs.

Algorithm 4 Cost Optimization for Parking Slot

```
for Extract $CP.LIST$ do
  while Get parking slot details form $sl.LIST$ do
    Calculate $\Upsilon_{max} = \{y(P) + z(C)\}$
    Calculate $\Upsilon_{avg} = y(P_{avg}) + z(C)$
  end while
  Compare $\Upsilon_{avg}$, $\Upsilon_{max}$
  Scanning each CPs
  for Each CPs in $CP.LIST$ do
    if $(1 \leq sl_n^{curr} \leq sl_n$ then
      $\Upsilon_{op} = \begin{Bmatrix} \Upsilon_{avg} \\ \Upsilon_{max} \end{Bmatrix}$
      Select Lowest value of $\Upsilon_{op}$, where $(0 < \Upsilon_{op} \leq 1)$
    else if $(1 \leq sl_n^{curr} \leq \frac{n}{2}$ then
      $\Upsilon_{avg} = y(P_{avg}) + z(C)$
      Compare $\Upsilon_{avg}$ for $LIST.sl_1^{\frac{n}{2}}$
      Select Lowest value of $\Upsilon_{avg}$
    else
      Select available $sl$ with $\Upsilon_{max}$
    end if
  end for
end for
```

straight away drive their vehicles to CPs and walk down to WP, if L-AVP is not involved.

E. TOTAL TRAVEL TIME

For L-AVP the outbound trip time for a single vehicle can be taken as T_{lavp}^{out}, which is computed by adding the time required for the vehicle to arrive at the drop-off spot and the time required by the driver to cover the distance between drop-off spot and desired destination. The relationship for T_{lavp}^{out} is defined as follows;

$$T_{lavp}^{out} = \frac{D_{v,d}}{S_v} + \frac{D_{d,w}}{S_h} \tag{9}$$

For the comparison purposes we use the benchmark case T_{bck}^{out}, where the total time is calculated as vehicle to arrive at the car park and the time required by the driver to cover the distance between car desired destination. The relationship for T_{bck}^{out} is defined as follows;

$$T_{bck}^{out} = \frac{D_{v,cps}}{S_v} + \frac{D_{cps,w}}{S_h} \tag{10}$$

As the value of S_v larger in comparison to S_h (e.g., 13.9-14.9 m/s vs 1.5-2.0 m/s). These are the values of $D_{d,w}$ and $D_{cps,w}$, which illustrates how optimal L-AVP is. The suggestions for D/P shows the efficacy of the L-AVP. The capacity of CPs is dependent on size & population of the city. Busy and congested areas need CPs with a larger capacity as compared to small cities, a detailed demonstration is given in Fig. 7. When higher number of D/P spots are deployed, there is a higher possibility to locate $d \in \mathcal{D}$ to hold $D_{d,w} < D_{cps,w}$.

The inbound time for L-AVP can be defined as;

$$T_{lavp}^{in} = \frac{D_{cps,x}}{S_v} + \frac{D_{w,p}}{S_h} \tag{11}$$

where, $D_{cps,x}$ is the distance from CPs to destination and being covered by AV. The user needs to walk down to pick-up spot $D_{w,p}$. The L-AVP tends to select the nearest pick-up spot to WP. As speed of human S_h is much lower than speed of vehicle S_v. To minimize the waiting time for user and AV at pick-up spot, their distance and speed must be synchronized.

Similarly, the benchmark for inbound trip can be presented as;

$$T_{bck}^{in} = \frac{D_{cps,x}}{S_v} + \frac{D_{w,cps}}{S_h} \tag{12}$$

For the benchmark, the AV needs to travel from its designated slot to CPs entrance $D_{cps,x}$ with the speed of S_v. It takes quite less time due to speed of vehicle. The user travels from WP to CPs $D_{w,cps}$. The CPs being deployed remotely, this distance is usually long and user needs to use some public transport or walk a long way to CPs from WP. It costs the user an extra amount or time or sometimes both to travel between WP and CPs.

F. TOTAL JOURNEY COST

The total journey cost of an AV in L-AVP includes, the fuel consumption for that specific AV from current location to drop-off spot $P_{v,d}$ then cost of fuel from drop-off spot to CPs $P_{d,cps}$ and optimized parking fee is obtained by Υ_{op};

$$J_{lavp-out} = \overbrace{P_{v,d} + P_{d,cps}} + \Upsilon_{op} \tag{13}$$

For benchmark total journey cost can be defined as;

$$J_{bck-out} = \overbrace{P_{v,cps} + U_{cps,w}} + \Upsilon_{max} \tag{14}$$

Here, $U_{cps,w}$ is the cost of user travelling from CPs to WP. $P_{v,cps}$ is the cost of AV from current location to CPs. While, Υ_{max} the cost of parking.

The consumption of fuel is directly proportional to the distance covered by vehicle. In case of L-AVP, $D_{v,d}+D_{d,cps}$ is traversed. While in benchmark, $D_{v,cps}$ is covered. If L-AVP relies on just one CPs and D/P, it will record a higher fuel consumption. As AV must be dropped at designed D/P and parked in defined CPs. In case of multiple D/P and CPs, AV will select the nearest D/P to WP and nearest CPs to D/P. It will result in less fuel consumption for AV and less walking distance for users.

IV. PERFORMANCE EVALUATION

A. SIMULATION SET-UP

The presented case study is carried out with Opportunistic Network Environment (ONE) [33]. The ONE is a Java-based simulator. The scenario is with an area of 4500×3400 m^2, shown as down town area of Helsinki city in Finland. The case study present 300 AV running at a speed range of $[30 \sim 50]$

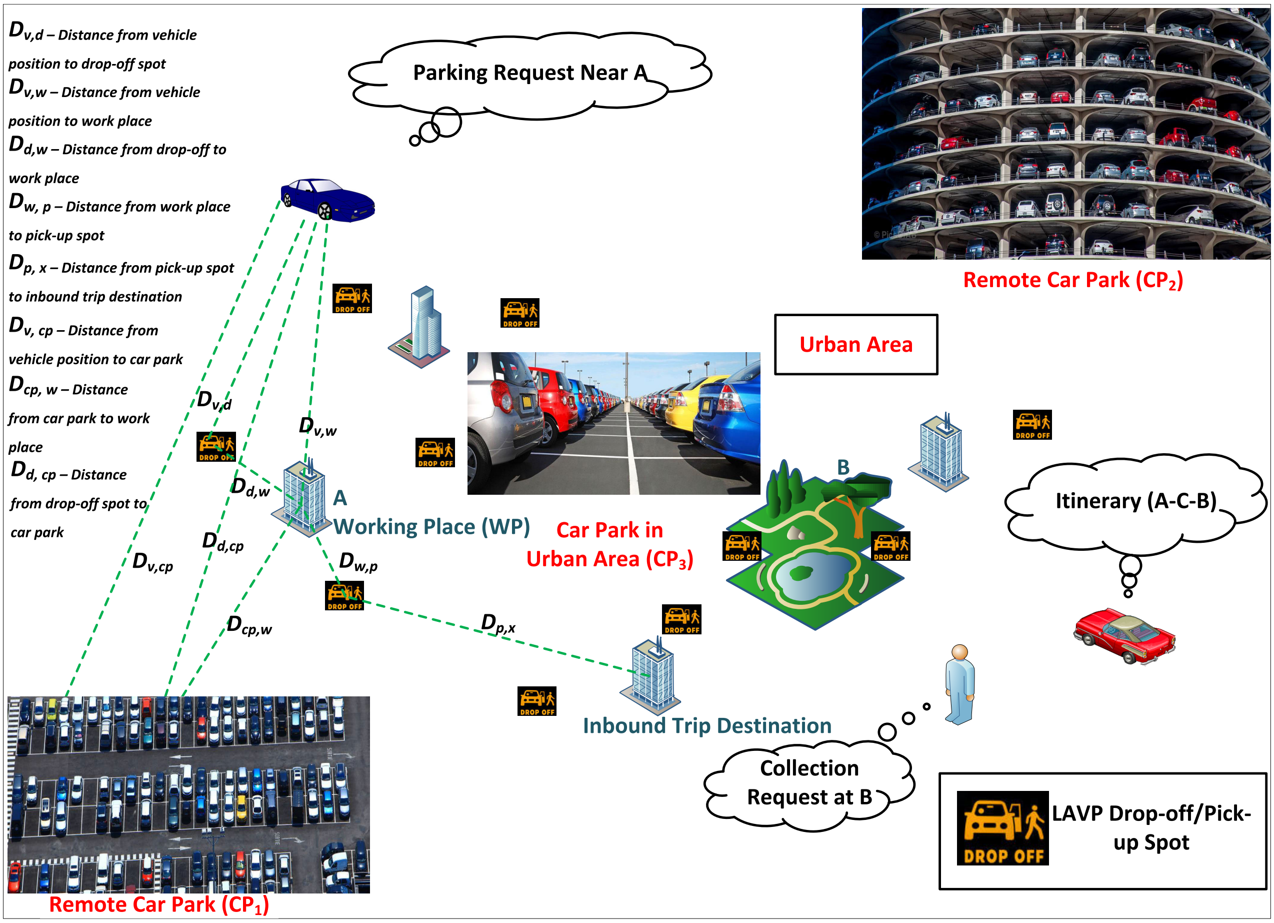
$D_{v,d}$ – Distance from vehicle position to drop-off spot
$D_{v,w}$ – Distance from vehicle position to work place
$D_{d,w}$ – Distance from drop-off to work place
$D_{w,p}$ – Distance from work place to pick-up spot
$D_{p,x}$ – Distance from pick-up spot to inbound trip destination
$D_{v,cp}$ – Distance from vehicle position to car park
$D_{cp,w}$ – Distance from car park to work place
$D_{d,cp}$ – Distance from drop-off spot to car park
Parking Request Near A
Remote Car Park (CP_2)
Urban Area
$D_{v,d}$
$D_{v,w}$
$D_{d,w}$
A
Working Place (WP)
Car Park in Urban Area (CP_3)
B
Itinerary (A-C-B)
$D_{d,cp}$
$D_{v,cp}$
$D_{w,p}$
$D_{p,x}$
$D_{cp,w}$
Inbound Trip Destination
Collection Request at B
DROP OFF
LAVP Drop-off/Pick-up Spot
Remote Car Park (CP_1)

FIGURE 7. Broad-view design for long-range autonomous valet parking.

TABLE 2. Simulation parameters.

Simulation Tool	Opportunistic Network Emulator (ONE)
Simulation Time	12 hours
City Map	Helsinki, Finland
Area	4500 x 3400 m^2
No. of AVs	300
Deployed Drop-off Spots	15
Deployed Car Parks	5
Drop-off Request Time	3600s
Working period at Work Place	7200s

km/h are deployed in the network. This case study deploys a total of 5 CPs (each CPs with 60 slots by default) and 15 D/Ps as depicted in Fig. 8. By default, the time for users to start requesting for drop-off spot is 3600s while 7200s is set as working period. The simulation has been carried out for 12 hours, defined in Table 2. The terminologies used in simulation are defined as follows;

- **Walking Distance (WD)** This is the distance covered by a user from drop-off spot to WP. This distance depends on the number of drop-off spots deployed inside the city center. The more the number of spots will lead to pick-up/drop people closer to WP.
- **Parking Cost (PC)** This is the actual parking fee C that a user is charged when parking their AV. The parking fee also depends on the number of hours an AV is parked for.
- **Total Fuel Consumption (TFC)** The TFC is the amount of fuel consumed per trip. A trip can be defined as leaving the drop-off spot. Then its arrival at designated CPs and travelling back to pick-up spot. The TFC can be minimized by selecting the shortest path to CPs or by

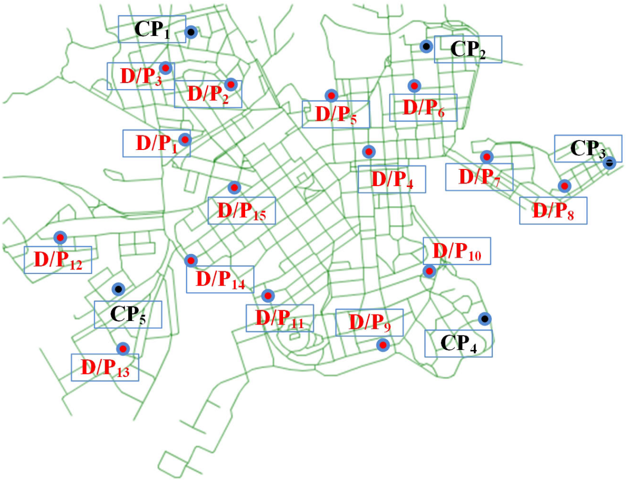

FIGURE 8. **Simulation scenario.**

selecting nearest CPs. Usually there is a trade-off between CPs and TFC. When former decreases, the latter increase and vice versa.

- **Average Travel Time (ATT)** The ATT involves two factors; one computed from user's perspective and the other from vehicle's perspective. It involves computation of time needed by AV to travel from a drop-off spot to CPs and time needed by driver to travel from drop-off spot to the desired location. It can be given by $\left\{\frac{D_{v,d}+D_{d,cps}+D_{cps,x}}{S_v}+\frac{D_{d,w}+D_{w,p}}{S_h}\right\}$. Here, $\frac{D_{v,d}+D_{d,cps}+D_{cps,p}}{S_v}$ is the time taken by AV while $\frac{D_{d,w}+D_{w,p}}{S_h}$ is the time taken by user in each trip.
- **Total Distance Covered (TDC)** It indicates the distance covered by AV and user in each trip. Where a trip start from drop-off spot to CPs for AV and to WP for user and similarly the inward trip to pick-up spot for both user and AV. Where distance covered by AV and user can be represented by $D_{v,d} + D_{d,cps} + D_{cps,x}$ and $D_{d,w} + D_{w,p}$ respectively. Similarly, for benchmark it will be $D_{v,cps}$ and $D_{cps,w}$ for AV and user respectively.

Fig. 9 shows the walking distance travelled by user from drop-off spot towards WP in L-AVP. The distance is compared against the number of drop-off spots deployed and results are compared for a total of 15 drop-off spots. The result shows that there is a significant decrease in walking distance from drop-off spot to WP with increase in number of drop-off spots deployed. The walking distance may vary depending on structure of the city center but overall walking distance from D/P to WP will be decreased with the increase of D/P spots. The walking distance can also be minimized by deploying D/P spots near WP but that can effect the flow of traffic. Fig. 10 provides the comparison between the distance covered by user as well as AV. The covered distance is compared concurrently against the drop-off spots and CPs. The increase in spots and CPs showed a prominent change in decrease of distance covered by both AV and user.

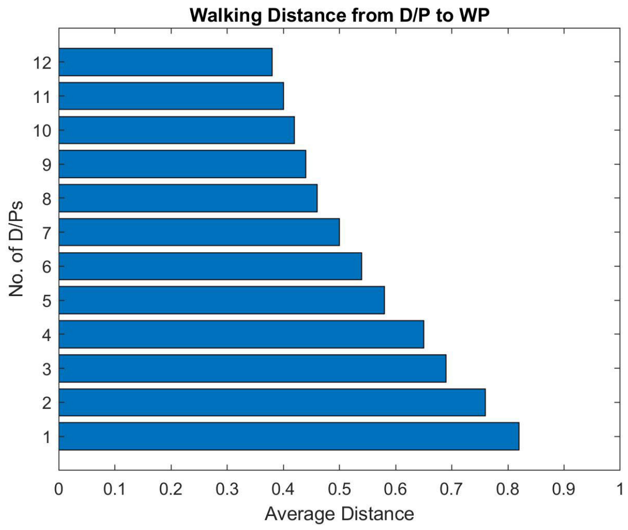

FIGURE 9. **Walking distance comparison for various D/P.**

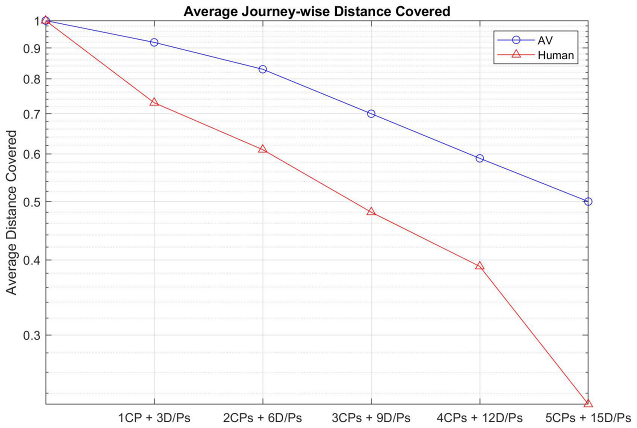

FIGURE 10. **Average distance covered.**

The more number of drop-off, pick-up spot and CPs, smaller the overall travel distance meaning that they are easily accessible. Similarly, it decreases the walking distance in L-AVP as well. The CPs away from city center are usually less costly as compare to CPs near the city center. One may pay lower parking fee but has to travel a larger distance to reach the CPs. The Fig. 11 represents the cost of parking for both benchmark and AVPark against the number of CPs deployed. The parking cost depends on the availability of parking slots. The parking cost of all the CPs are compared through optimization equation and CPs with the minimized cost is selected. The CPs are given with different prices, depending on how far they are from city center. The optimal cost of parking has been achieved through the selection of nearest D/P and then scanning for the lowest parking price in combination with selection shortest path to CPs.

The cost against different values of P as fuel consumed and C as parking cost has been shown in Fig.13. The results are being shown for parking hours between 9am-5pm. The total cost progressively decreases with an increase in D/Ps

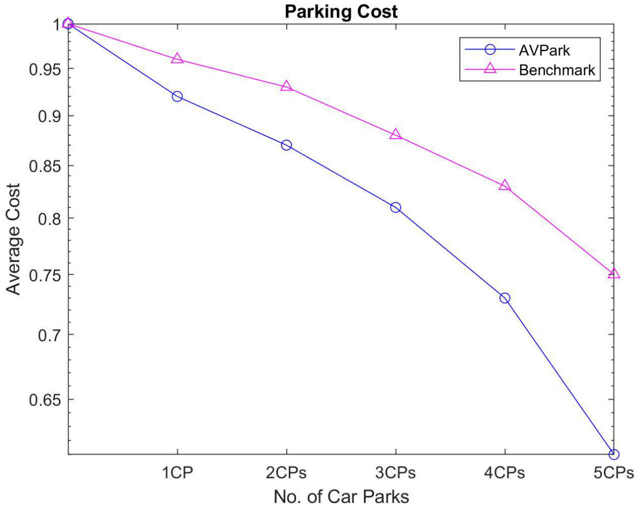

FIGURE 11. **Parking cost.**

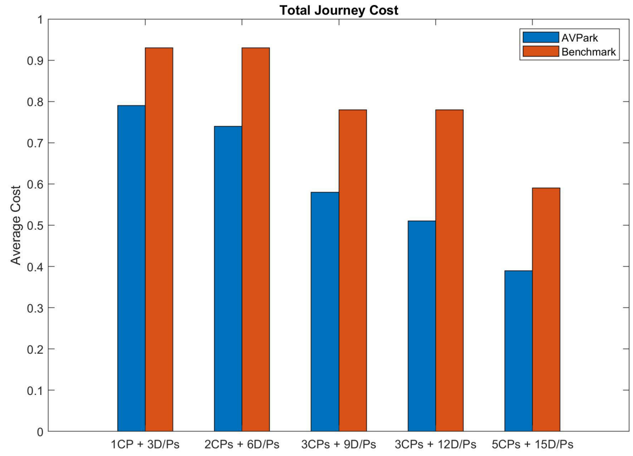

FIGURE 12. **Total journey cost.**

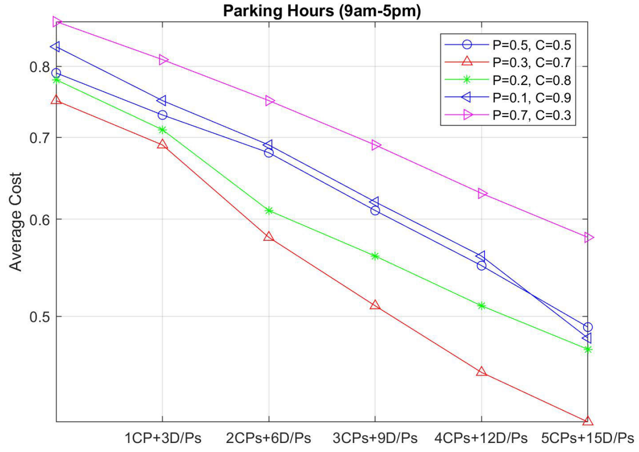

FIGURE 13. **Average cost with respect to *P* & *C* Values.**

and CPs. The value of P as 0.3 and C as 0.7 offers the lowest cost in the simulated scenario. A variation in cost with respect to different values of P and C can also be observed. The Fig. 12 depicts the total cost of the outward and inward trip, also called as total journey. TJC in benchmark includes the parking cost and journey cost from CPs to WP. The travel between CPs and WP may be by bus, train or by walk. The TJC of AVPark and benchmark are compared in Fig. 12. The Fig.13 represents total cost with respect to different values of P as fuel consumption and C as parking price. The most optimal value is achieved by $P = 0.3$ and $C = 0.7$. The Fig. 14 has analyzed ATT with respect to number of drop-off, pick-up spots and CPs. The ATT represents the journey time starting from the time AV requests for parking spot. Until the AV is parked in CPs and the time AV requests for pick-up spot till AV is picked by the driver. Results show that ATT decreases with increase in D/P and CPs simultaneously. The higher number of D/Ps means, AV can easily find a nearby D/P and results in minimizing ATT. The ATT of AVPark is compared with benchmark. In the benchmark the CPs entrance is considered as drop spot and pick up spot simultaneously.

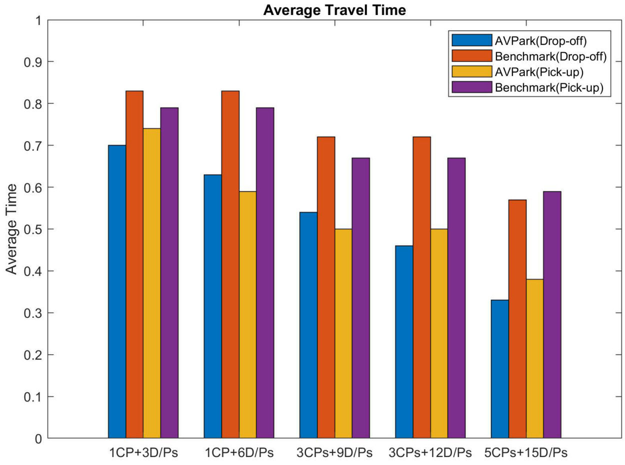

FIGURE 14. **Average travel time.**

V. CONCLUSION

This paper proposed a novel technique for optimizing reservation process and minimizing cost of using parking areas, called AVPark. A new model for AVP has been presented in this article. The L-AVP for outbound trip, uses the nearest D/P to drop user at a convenient location near WP. Similarly, for the inbound trip the most convenient D/P is selected and AV is picked by the user at that spot. The results of the novel technique were compared with the benchmark and the proposed technique has achieved improvement in minimizing the time required for travelling, parking cost, fuel consumption and distance to be travelled by AV. In future, Integration of AVP with edge computing and cloud to support IoT services will be of great interest. The need of IoT applications that require location awareness, real-time and low-latency responses, core network bandwidth load management, data security management, and IoT power consumption management can be the motivation. Real-time video analytics for low-latency decision making is an IoT application area which is seen to have real benefits when the processing is done at the edge nodes. This type of analytics will be needed for making self-driving cars and augmented reality.

REFERENCES

[1] "Transport statistics Great Britain," Dept. Transp., Great Britain, Tech. Rep. 2017. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/661933/tsgb-2017-report-summaries.pdf

[2] (2019). *Justpark*. [Online]. Available: https://www.justpark.com

[3] S. Gallivan, "IBM global parking survey: Drivers share worldwide parking woes technical report," New York, NY, USA, IBM, Tech. Rep. 2011.

[4] A. Klappenecker, H. Lee, and J. L. Welch, "Finding available parking spaces made easy," *Ad Hoc Netw.*, vol. 12, pp. 243–249, Jan. 2014.

[5] E. Inci, "A review of the economics of parking," *Econ. Transp.*, vol. 4, nos. 1–2, pp. 50–63, Jun. 2015.

[6] R. Arnott, E. Inci, and J. Rowse, "Downtown curbside parking capacity," *J. Urban Econ.*, vol. 86, pp. 83–97, Mar. 2015.

[7] N. Kafle, B. Zou, and J. Lin, "Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery," *Transp. Res. B, Methodol.*, vol. 99, pp. 62–82, May 2017.

[8] A. Y. Lam, J. Q. James, Y. Hou, and V. O. Li, "Coordinated autonomous vehicle parking for vehicle-to-grid services: Formulation and distributed algorithm," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4356–4366, Jan. 2017.

[9] D. Z. Wang, I. Posner, and P. Newman, "Model-free detection and tracking of dynamic objects with 2D lidar," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 1039–1063, Jun. 2015.

[10] U. Schwesinger *et al.*, "Automated valet parking and charging for E-mobility," in *Proc. IEEE Intell Vehicles Symp. (IV)*, Jun. 2016, pp. 157–164.

[11] M. Khalid, Y. Cao, X. Zhang, C. Han, L. Peng, N. Aslam, and N. Ahmad, "Towards autonomy: Cost-effective scheduling for long-range autonomous valet parking (LAVP)," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[12] A. O. Kotb, Y.-C. Shen, X. Zhu, and Y. Huang, "IParker—A new smart car-parking system based on dynamic resource allocation and pricing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2637–2647, Sep. 2016.

[13] M.-F. Tsai, Y. C. Kiong, and A. Sinn, "Smart service relying on Internet of Things technology in parking systems," *J. Supercomput.*, vol. 74, no. 9, pp. 4315–4338, 2018.

[14] L. Zhang and Y. Li, "Optimal management for parking-lot electric vehicle charging by two-stage approximate dynamic programming," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1722–1730, Jul. 2017.

[15] T. Wreiner, I. Mårtensson, O. Arnell, N. Gonzàlez, S. Holmlid, and F. Segelström, "Exploring service blueprints for multiple actors: A case study of car parking services," in *Proc. 1st Nordic Conf. Service Design*, Sep. 2012, pp. 213–223.

[16] M. Khalid, Z. Ullah, N. Ahmad, M. Arshad, B. Jan, Y. Cao, and A. Adnan, "A survey of routing issues and associated protocols in underwater wireless sensor networks," *J. Sensors*, vol. 17, Sep. 2017, Art. no. 7539751.

[17] A. Bagula, L. Castelli, and M. Zennaro, "On the design of smart parking networks in the smart cities: An optimal sensor placement model," *Sensors*, vol. 15, no. 7, pp. 15443–15467, Jul. 2015.

[18] M. S. Kuran, A. Carneiro Viana, L. Iannone, D. Kofman, G. Mermoud, and J. P. Vasseur, "A smart parking lot management system for scheduling the recharging of electric vehicles," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 2942–2953, Nov. 2015.

[19] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. S. Shen, "Privacy-preserving smart parking navigation supporting efficient driving guidance retrieval," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6504–6517, Jul. 2018.

[20] Y. Geng and C. G. Cassandras, "New 'smart parking' system based on resource allocation and reservations," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1129–1139, Sep. 2013.

[21] W. Griggs, J. Y. Yu, F. Wirth, and F. Häusler, and R. Shorten, "On the design of campus parking systems with qos guarantees," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1428–1437, May 2016.

[22] M. Chirca, R. Chapuis, and R. Lenain, "Autonomous valet parking system architecture," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2015, pp. 2619–2624.

[23] K. Min, J. Choi, H. Kim, and H. Myung, "Design and implementation of path generation algorithm for controlling autonomous driving and parking," in *Proc. 12th Int. Conf. Control, Automat. Syst.*, Oct. 2012, pp. 956–959.

[24] A. Broggi, E. Cardarelli, S. Cattani, P. Medici, and M. Sabbatelli, "Vehicle detection for autonomous parking using a soft-cascade AdaBoost classifier," in *Proc. IEEE Intell. Vehicles Symp. Proc.*, Jun. 2014, pp. 912–917.

[25] K. An, J. Choi, and D. Kwak, "Automatic valet parking system incorporating a nomadic device and parking servers," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2011, pp. 111–112.

[26] S. E. Shladover, "Connected and automated vehicle systems: Introduction and overview," *J. Intell. Transp. Syst.*, vol. 22, pp. 190–200, May 2018.

[27] S. Klaudt, A. Zlocki, and L. Eckstein, "A-priori map information and path planning for automated valet-parking," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1770–1775.

[28] M. Y. Aalsalem, W. Z. Khan, and K. M. Dhabbah, "An automated vehicle parking monitoring and management system using ANPR cameras," in *Proc. 17th Int. Conf. Adv. Commun. Technol. (ICACT)*, Jul. 2015, pp. 706–710.

[29] E. Wu, J. Sahoo, C. Liu, M. Jin, and S. Lin, "Agile urban parking recommendation service for intelligent vehicular guiding system," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 1, pp. 35–49, Mar. 2014.

[30] A. Broggi, E. Cardarelli, S. Cattani, P. Medici, and M. Sabbatelli, "Vehicle detection for autonomous parking using a Soft-Cascade AdaBoost classifier," in *Proc. IEEE Intell. Vehicles Symp. Proc.*, Jun. 2014, pp. 912–917.

[31] M. Khalid, Y. Cao, N. Aslam, C. Suthaputchakun, M. Arshad, and W. Khalid, "Optimized pricing & scheduling model for long range autonomous valet parking," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2018, pp. 65–70.

[32] E. K. Nam and R. Giannelli, *Fuel Consumption Modeling of Conventional and Advanced Technology Vehicles in the Physical Emission Rate Estimator (PERE)*. BiblioGov, USA, 2013.

[33] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. 2nd Int. Conf. Simulation Tools Techn.*, Mar. 2009, p. 55.

MUHAMMAD KHALID received the M.S. degree in computer science from the Institute of Management Sciences, Peshawar, Pakistan. He is currently pursuing the Ph.D. degree with Northumbria University, Newcastle Upon Tyne, U.K. His research interests include EV charging and scheduling, the Internet of Things, wireless sensor networks, and autonomous valet parking.

YUE CAO received the Ph.D. degree from the Institute for Communication Systems (ICS), University of Surrey, U.K., in 2013. He was the Research Fellow at ICS, University of Surrey; a Lecturer and a Senior Lecturer with the Department of Computer and Information Sciences, Northumbria University, U.K.; and has been the International Lecturer with the School of Computing and Communications, Lancaster University, U.K. His research interest includes intelligent transport systems. He is the Associate Editor of the IEEE Access, *KSII Transactions on Internet and Information Systems*, *IGI Global International Journal of Vehicular Telematics and Infotainment Systems*, and *EURASIP Journal on Wireless Communication and Networking* (Springer).

NAUMAN ASLAM received the Ph.D. degree in engineering mathematics from Dalhousie University, Halifax, NS, Canada, in 2008. He was an Assistant Professor with Dalhousie University, until 2011. He is currently a Reader with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, U.K. His research interests include wireless ad hoc and sensor networks, fault tolerant and reliable communication, and remote health monitoring application.

MOHSIN RAZA received the B.S. (Hons.) and M.S. degrees in electronic engineering from Mohammad Ali Jinnah University, Islamabad, Pakistan, and the Ph.D. degree from the Department of Mathematics, Physics and Electrical Engineering, Northumbria University, Newcastle upon Tyne, U.K. He was a Lecturer with the Department of Electronic Engineering, Mohammad Ali Jinnah University, from 2010 to 2015, and a Hardware Support Engineer with USS, in 2009. He is currently a Postdoctoral Fellow with Middlesex University, London. His research interests include wireless communications, future networks, device-to-device communications, and multi-hop communications for emergency, feedback and monitoring systems, and wireless sensor networks.

ALUN MOON is currently a Senior Lecturer with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, U.K. His research interests include wireless ad hoc and sensor networks, machine learning, and image processing.

HUAN ZHOU received the Ph.D. degree from the Department of Control Science and Engineering, Zhejiang University. He was a Visiting Scholar with Temple University, from 2012 to 2013, and a CSC supported Postdoctoral Fellow with The University of British Columbia, from 2016 to 2017. He is currently a school-hired Professor with the College of Computer and Information Technology, China Three Gorges University. His research interests include mobile social networks, VANETs, opportunistic mobile networks, and wireless sensor networks. He is serving as an Associate Editor for IEEE Access, and the Local Arrangement Co-Chairs of I-SPAN 2018.