A Quality of Service-Aware Least Utilization First Algorithm for Smart On-Street Parking Management

Sanjaya Kumar Panda, *Senior Member*, *IEEE*
Department of Computer Science and Engineering
National Institute of Technology Warangal
Warangal - 506004, Telangana, India
sanjaya@nitw.ac.in

Dhiraj Shinde
Department of Computer Science and Engineering
National Institute of Technology Warangal
Warangal - 506004, Telangana, India
dhirajshinde.nitw@gmail.com

***Abstract*—The rapid growth in the number of vehicles and limited parking slots has made on-street parking management an increasingly complex challenge. Intelligent transportation systems (ITS) have emerged as a key solution, enhancing the efficiency of smart on-street parking management by delivering real-time information on slot availability, optimizing space utilization, and significantly reducing the time drivers spend searching for parking. Researchers have developed various pricing strategies to map the parking requests to the parking slots, maximizing revenue generation. However, most strategies only consider the provider's perspective to improve the utilization of parking slots without considering the user's perspective, leading to dissatisfaction among the users. Therefore, we model the user's perspective using the quality of service (QoS) and introduce a QoS-aware least utilization first (LUF) algorithm for smart on-street parking management. LUF aims to maximize revenue generation through a dynamic pricing strategy without compromising user satisfaction. It satisfies the QoS of the parking requests submitted by the users, thereby selecting the less-utilized parking slots. The performance of LUF is evaluated by taking the Seattle city dataset and compared using three algorithms, namely first come, first serve (FCFS), round robin (RR), and highest price first (HPF), in terms of number of accepted requests (R_{acc}), number of rejected requests (R_{rej}), average price (AV) and revenue (RV). The simulation results across ten areas of the Seattle city dataset show that LUF outperforms in all the performance metrics compared to FCFS, RR, and HPF.**

***Index Terms*—Intelligent Transportation Systems, Smart On-Street Parking Management, Quality of Service, Least Utilization First, Dynamic Pricing Strategy, Seattle City, Revenue Generation.**

I. Introduction

On-street parking management is a significant challenge for urban cities, where growing vehicle ownership and restricted parking infrastructure create a complex optimization problem [1], [2]. Improper parking management leads to traffic congestion, fuel wastage, higher carbon emissions, and compromised quality of life for urban residents [1]. For instance, it is estimated that drivers looking for parking slots cause at least 30% of traffic congestion in urban cities [2]. Consequently, congestion prolongs the travel time for drivers who do not seek parking and further reduces the available road space. On the other hand, as per the Precedence research [3], the United States (US) smart parking systems market exceeds US dollar (USD) 2.47 billion in 2024, and it is expected to reach 17.78 billion by 2034 with a compound annual growth rate (CAGR) of 21.82% from 2025 to 2034. Similarly, the global smart parking systems market was estimated at USD 9.15 billion in 2024, and it is expected to reach 64.50 billion by 2034 with a CAGR of 21.57% from 2025 to 2034 [3]. ITS presents a viable solution for effective on-street parking management by providing real-time data on parking slot availability, optimizing space utilization, streamlining revenue tracking, prioritizing parking allocation, and minimizing drivers' time searching for parking [4]–[10]. Therefore, many researchers have developed various algorithms to maximize parking slot availability, space utilization, and revenue, thereby minimizing search time [4], [11], [12]. Specifically, researchers have developed various pricing strategies to map between the parking requests and the parking slots to maximize revenue generation. However, such strategies must consider both the service provider's and the user's perspectives to enhance parking slot utilization and user satisfaction [13].

Efficient on-street parking management is vital for generating revenue that can be reinvested in infrastructure development and improving public transportation systems [1], [2]. However, a common challenge is the uneven distribution of parking demand, where high-demand areas experience full occupancy while adjacent areas remain underutilized. As a result, parking requests in high-demand areas are often disregarded or not fulfilled as per choice, leading to reduced QoS and increased dissatisfaction among drivers [4], [11], [12]. On the other hand, pricing strategies are vital in the on-street parking management. Such strategies can be broadly categorized into fixed and dynamic. In a fixed pricing strategy, all parking requests are charged a uniform rate regardless of parking slot location and/or current space utilization, often resulting in suboptimal revenue generation [14]. Alternatively, such a strategy fails to generate revenue on the dynamic nature of parking slot demand patterns, resulting in inefficient utilization of parking slots. In contrast, a dynamic pricing strategy determines charges based on various factors such as parking slot location, current utilization levels, demand patterns, etc. [11], [15]. It is also regarded as a future-proof parking management strategy, providing drivers with real-time pricing information based on current demand patterns [15]. However, the efficiency of the dynamic pricing strategy heavily depends on the underlying mapping algorithm that

distributes available parking slots to incoming parking requests by looking into the user's and provider's perspectives.

This paper addresses the mapping problem of n parking requests to m parking slots to maximize the R_{acc} and the RV, and introduces a QoS-aware LUF algorithm for smart on-street parking management. LUF algorithm assigns the parking requests to the parking slots by keeping both the user's and provider's perspectives. Specifically, once a parking request is received, the LUF algorithm checks the corresponding QoS. The QoS is high if the parking request is specified with a particular location. Otherwise, it is low. Then, the LUF algorithm determines the least utilized parking slot and assigns the parking request to the corresponding parking slot. Moreover, the LUF algorithm employs a dynamic pricing strategy with three zones, namely green, yellow, and red, based on the utilization rate of parking locations without compromising user satisfaction. We perform a series of simulation runs using ten distinct areas from the Seattle city dataset to assess the performance of the LUF algorithm. Then, we compare the simulation results of the LUF algorithm with three algorithms, FCFS, RR [16]–[19], and HPF, in terms of the four performance metrics, R_{acc}, R_{rej}, AV, and RV. Here, the FCFS algorithm sequentially assigns parking requests to the parking slots without considering the parking locations. The RR algorithm sequentially assigns parking requests to the parking slots by considering the utilization of parking locations without QoS. The HPF algorithm assigns the parking requests to the parking slots that generate maximum revenue. The comparison results show that the LUF algorithm outperforms three algorithms in four performance metrics, showing a balance between the QoS of parking requests, revenue generation, and utilization of parking slots.

The remainder of the paper is structured as follows. Section II reviews the related work and the gaps filled by the LUF algorithm. Section III outlines the system model and formally defines the problem. Section IV presents the LUF algorithm. Section V shows the datasets, simulation results, and discussion. Finally, Section VI concludes the paper and offers directions for future research.

II. Related Work

Efficient on-street parking management is quite challenging in many urban cities, exacerbated by the increased number of vehicles and the lack of parking slots. Many algorithms have been developed to fulfill various objectives, from parking slot management to dynamic pricing strategies. However, many existing algorithms fail to align with both user and provider perspectives, overlook QoS considerations, and are not well-suited for real-time decision-making.

Lei and Jasin [20] have presented real-time dynamic pricing for resource management to maximize revenues by developing two heuristics. However, their heuristics do not consider user preferences and QoS. Therefore, they are unsuitable for on-street parking scenarios, especially when real-time availability and user satisfaction hold key importance. Ahmed and Rahman [21] have proposed a blockchain-enabled architecture designed to support integrated smart parking systems. Their architecture aims to improve data integrity and security. However, it neither utilizes dynamic pricing nor considers the utilization of parking slots to maximize revenue generation and user satisfaction.

Mak [22] has shown the importance of operations management for smart city initiatives. They have focused on both the public and private sectors. They have mentioned transportation as one of the promising research domains. Bhatia and Sood [23] have mapped the large Internet of Things (IoT) applications and the fog resources to perform real-time data analysis by incorporating scheduling. They emphasize predictive load scheduling. However, their scheduling does not explicitly consider the utilization of fog resources. Qin et al. [24] have proposed a parking recommendation model based on traveler psychological behavior and various parking factors. However, their model fails to incorporate dynamic pricing based on the variation in demand. Zhu and Cai [25] have presented a multi-agent model to optimize installation configurations of electric vehicle charging piles in public garages. The model is introduced to show parking behavior. However, they have not considered the differences in the charges for parking slots.

Garra et al. [26] have discussed the privacy issues related to parking information, especially user profiles, and suggested a pay-by-phone system without creating the user profiles to preserve privacy. However, they have not shown any mapping between the parking request and the parking slots, thereby any pricing strategy. Saharan et al. [11] have investigated adaptive solutions like machine learning-based predictions to predict parking demand dynamically and enhance allocation efficiency. An et al. [27] have utilized deep learning techniques to improve the pricing strategy by using real-time traffic flow and parking occupancy. Saharan et al. [12] have developed the foggy-park framework based on dynamic pricing and multi-objective optimization for on-street parking. Their framework uses a non-dominated sorting genetic algorithm (NSGA) allocation to increase revenue. However, their framework does not explicitly consider the utilization of parking slots. The proposed algorithm, LUF, is similar to and differs from the existing works in the following ways. (1) Unlike foggy-park [12], the LUF algorithm considers the user preference in terms of QoS for assigning a parking slot. (2) Unlike foggy-park [12], the LUF algorithm considers the utilization of parking slots to select an appropriate parking slot for each parking request. (3) Unlike foggy-park [12], the LUF algorithm uses a dynamic pricing strategy with three zones to hike the revenue of parking slots based on the dynamic demands.

III. System Model and Problem Formulation

The system model considers an area (AR) that provides on-street parking management. Each AR is divided into multiple sub-areas (SAs), which are further divided into block faces (BFs). A BF may be further divided into street sides (SSs), which are further divided into parking lots (PLs). A PL may further be divided into parking slots (PSs). Alternatively, PSs

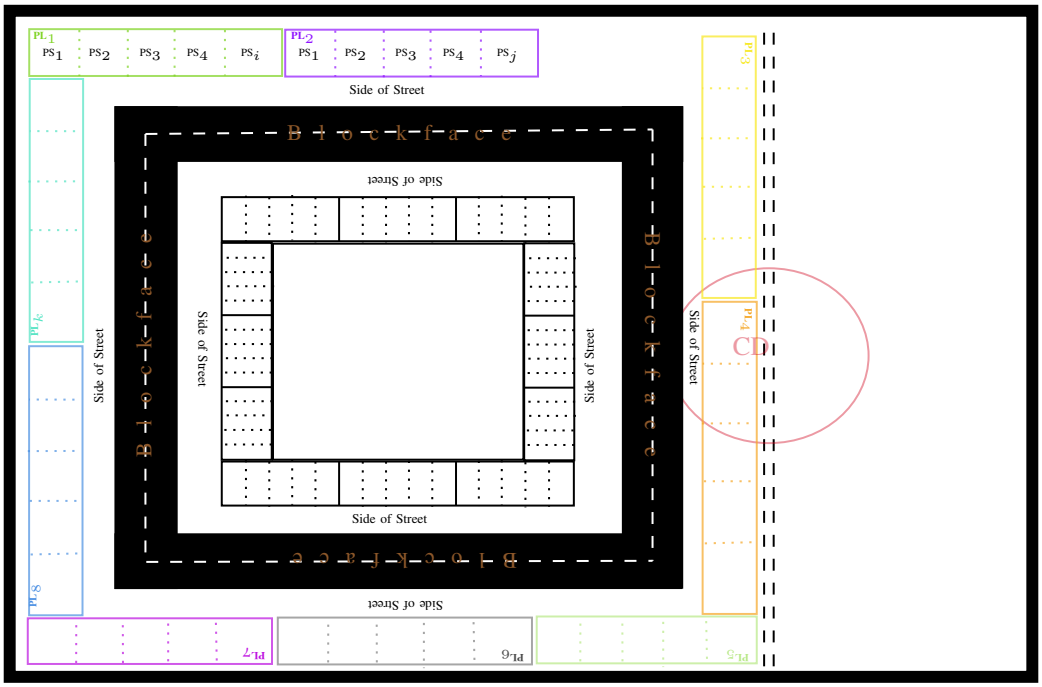

Fig. 1: A parking system with a single AR.

on one side of the street are grouped into a single PL. The system model follows a hierarchical structure, as shown in Fig. 1. A user submits a parking request through an Internet-based application. This request is captured by the computing device (CD) deployed by the provider within the AR. Note that the parking request can be submitted in two forms, i.e., with and without QoS. A parking request with QoS is associated with a specific BF, whereas a request without QoS does not specify a BF. Once the QoS is determined, the dynamic price is calculated based on the utilization of the BF. Then, the user must pay the price for assigning the PS for the specified duration.

Consider a set of n parking requests, $R = \{R_1, R_2, R_3, \ldots, R_n\}$, a set of m PSs, $PS = \{PS_1, PS_2, PS_3, \ldots, PS_m\}$ in an AR and $n >> m$. Each parking request, R_i, $1 \leq i \leq n$, is represented in a 4-tuple, <TS, AR, BF, D>, where TS denotes the arrival timestamp of the parking request and D denotes duration in time units. Note that the time of the day (TD) is determined as morning (M), afternoon (A), and evening (E) based on the TS. A global queue Q is maintained to keep the parking requests that arrived at CD based on the timestamp. If a parking request is specified with the BF, then we consider the request as with QoS. On the other hand, if it is not specified, then we consider the request to be without QoS. Similarly, each PS_j, $1 \leq j \leq m$ is associated with attributes, such as its current occupancy status and the base pricing (BP) according to its AR and/or SA. The problem is to map the parking requests to available PSs in such a manner that the following objectives are satisfied. (1) Maximize the R_{acc} and RV (2) Minimize the R_{rej} and AV. Mathematically,

$$\text{Maximize } R_{acc} \text{ and RV} \tag{1}$$

$$\text{Minimize } R_{rej} \text{ and AV} \tag{2}$$

The problem is restricted to the following constraints. (1) The mapping must ensure that no two parking requests occupy the same PS at the same time. (2) If two parking requests arrive at the same time, one with QoS and one without QoS, then they are processed in the order of QoS and then without QoS. In other cases, they are processed in chronological order.

IV. Proposed Algorithm

We propose a QoS-aware LUF algorithm for smart on-street parking management. The objective is to maximize the R_{acc} and RV, and minimize the R_{rej} and AV. The LUF algorithm makes the trade-off between RV and user dissatisfaction. We initialize the total number of PSs (TP) and the number of occupied PSs (OP) for each BF. The LUF algorithm picks a parking request from the Q and determines its QoS, as shown in Algorithm 1. Here, QoS is determined based on the BF that the user requests. If the parking request is submitted with QoS, then the utilization of BF is calculated, and the dynamic price is calculated. The utilization of the BF (UZ) is the ratio between the OP and the TP. Mathematically, $UZ = \frac{OP}{TP}$. Then, the LUF algorithm orders the available BFs from lowest to highest UZ to increase the utilization of underutilized BFs. On the other hand, if the parking request is submitted without QoS, then the least utilized BF is identified, and the dynamic price is calculated accordingly. For this, the LUF algorithm introduces three zones: green, yellow, and red. In the green zone, the user pays the base price (BP) as the utilization of BF is less than or equal to τ_1. The BP is taken based on the TD and the SA. This zone does not charge users more as there is sufficient availability of PSs. Note that this

Algorithm 1 LUF

Input(s): Q, R, PS, n, m, TS, AR, BF, SS, D, BP, τ_1 and τ_2

Output(s): R_{acc}, R_{rej}, AV and RV

for each BF_i in AR **do**
  if TP - $OP > 0$ **then**
    Calculate UZ = $\frac{OP}{TP}$
  end if
end for
Sort the queue Q based on QoS
for each request R_i in Q **do**
  Determine ID, AR, BF and D
  for each BF_i in AR **do**
    if BF of request R_i matches with BF_i **then**
      if TP - $OP > 0$ **then**
        OP = OP + 1
        Determine TD based on TS
        FP = $DYNAMIC\text{-}PRICE(TD, UZ, BP, \tau_1$ and $\tau_2)$
        R_{acc} = R_{acc} + 1 ▷ Assign request R_i to the available PS
        Update UZ
        break
      end if
    end if
  end for
  Sort the BFs based on their UZ
  if BF of request R_i is not specified **then**
    Determine the BF with the least UZ
    if TP - $OP > 0$ **then**
      OP = OP + 1
      Determine TD based on TS
      FP = $DYNAMIC\text{-}PRICE(TD, UZ, BP, \tau_1$ and $\tau_2)$
      R_{acc} = R_{acc} + 1
      Update UZ
      break
    end if
  end if
  if no slot is found **then**
    R_{rej} = R_{rej} + 1 ▷ Reject request R_i
  end if
end for
Calculate AV using Eq. (6)
Calculate RV using Eq. (7)

zone is considered to deal with low-demand hours without charging any dynamic price component. The final price is calculated as the base price. Mathematically,

$$FP = BP \tag{3}$$

In the yellow zone, the user pays the BP as well as the dynamic price as the UZ is more than τ_1 and less than or equal to τ_2. This zone applies a dynamic factor for moderately

Procedure 1 $DYNAMIC\text{-}PRICE(TD, UZ, BP, \tau_1$ and $\tau_2)$

Find the BP for given TD
if $UZ \leq \tau_1$ **then**
  Calculate FP using Eq. (3)
  return FP
else if $UZ \leq \tau_2$ **then**
  Calculate FP using Eq. (4)
  return FP
else if $UZ > \tau_2$ **then**
  Calculate FP using Eq. (5)
  return FP
end if

raising the price and showing the increasing demand. Moreover, this zone prevents sudden price hikes, thus ensuring user satisfaction. The final price is calculated as follows.

$$FP = BP \times \sqrt{1 + UZ} \tag{4}$$

In the red zone, the user also pays the BP as well as the dynamic price as the UZ is more than τ_2. This zone applies a much higher dynamic factor (i.e., α) for aggressively raising the price and showing the huge demand. Moreover, this zone hikes prices to minimize congestion and assigns the PSs only to the more needy users. The final price is calculated as follows.

$$FP = BP \times (\sqrt{1 + UZ} + e^{UZ}) \tag{5}$$

The above pricing strategy ensures fair access to PSs while increasing RV and user satisfaction and reducing congestion. It is also shown in Procedure 1. Once the user pays the required price, one of the available PSs under the BF is assigned to the user for parking. Note that the LUF algorithm determines the dynamic price based on the real-time utilization rate of BFs, ensuring less congestion and proper distribution and utilization of the PSs. The AV is the ratio between the sum of the FP of the accepted requests and the R_{acc}. Mathematically,

$$AV = \frac{\sum FP}{R_{acc}} \tag{6}$$

The RV is the sum of the product of the FP and the D. It can be expressed as follows.

$$RV = \sum (FP \times D) \tag{7}$$

A. Illustration

We consider a typical illustration with 17 parking requests (R_1 to R_{17}) with QoS as shown in Table I, 15 PSs (PS_1 to PS_{15}) and an occupancy data as shown in Table II. This illustration presents a condition of moderate over-demand, which reflects typical congestion situations in urban cities. The current UZ of SSs of BFs is 50%, 80%, 80%, 81.81%, 75%, 66.67%, 85.71%, 60%, 85.71%, 83.34%, 87.50% and 33.34%, respectively. Here, we consider BP as 1 unit. The LUF algorithm first assigns the parking requests with QoS.

TABLE I: Parking requests and QoS

Parking Request	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}	R_{17}
TS	27-04-2025 10:10																
AR	AR_1																
BF	-	-	-	3	0	2	-	5	1	4	6	-	-	-	-	5	0
D (in minutes)	60	120	90	240	240	120	240	240	120	240	240	60	120	30	30	240	120
QoS	No	No	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes

TABLE II: Occupancy data

BF	0	0	1	1	2	3	3	4	4	5	5	6
SS	East	West	East	West	West	East	West	East	West	East	West	West
TP	2	5	5	11	4	3	7	5	7	6	8	3
OP	1	4	4	9	3	2	6	3	6	5	7	1

TABLE III: Comparison of four performance metrics for FCFS, RR, HPF, and LUF algorithms

Algorithm	n	m	R_{acc}	R_{rej}	AV	RV
FCFS	17	15	14	3	2.96	086.78
RR	17	15	14	3	2.96	086.75
HPF	17	15	15	2	2.85	090.45
LUF	17	15	15	2	2.85	110.69

For instance, R_4 is assigned to east side of BF_3, R_5 is assigned to east side of BF_0, R_6 is assigned to west side of BF_2, and so on. The current UZ of SSs of BFs is 100%, 100%, 100%, 81.81%, 100%, 100%, 85.71%, 80%, 85.71%, 100%, 100% and 66.67%, respectively.

Next, the LUF algorithm assigns the parking requests without QoS. For instance, R_1 is assigned to west side of BF_6, R_2 is assigned to east side of BF_4, R_3 is assigned to west side of BF_1, and so on. The current UZ of SSs of BFs is 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100%, 100% and 100%, respectively. However, R_{14} and R_{15} are rejected due to the unavailability of PSs. Therefore, R_{acc}, R_{rej}, AV, and RV are 15, 2, 2.85 (i.e., $\frac{42.70}{15}$), and 110.69. We compare the illustration results with three algorithms, FCFS, RR, and HPF, as shown in Table III. As seen from the table, the LUF algorithm outperforms all the other algorithms in R_{acc}, R_{rej}, AV, and RV. This improvement demonstrates the LUF algorithm's ability to gain maximum revenue by strategically mapping parking requests according to real-time utilization.

V. Simulation Results and Discussion

We conducted a series of simulation runs by taking ten ARs of the Seattle city dataset of 09-04-2024 [28]. The names of these areas are 12th avenue, 15th avenue E, commercial core, ballard, ballard locks, green lake, cherry hill, roosevelt, university district and first hill, and they are denoted as A to J, respectively. Table IV contains the initial setup of ten ARs in which US and RS represent the number of unique PSs and the number of remaining PSs (i.e., TP - OP). It is noteworthy to mention that the US is not the product of BF and SS, as SS may be in any of the cardinal and intercardinal directions, namely north, south, east, west, northeast, northwest, southeast, southwest, or any of their combinations. In

TABLE IV: Ten ARs of the Seattle city dataset

AR	BF	SS	US	TP	OP	RS	n	TD	BP
A	07	2	012	065	031	034	039	E	5.0
B	10	4	016	080	037	043	048	E	3.0
C	92	7	122	690	211	479	527	E	1.0 (FC), 1.5 (RL), 4.5 (WF)
D	34	8	062	527	181	364	401	E	5.5 (CR), 5.0 (ED)
E	01	2	002	072	012	060	066	A	2.0
F	20	8	030	136	050	086	095	M	2.0
G	04	3	006	056	006	050	055	M	2.0
H	08	4	010	069	021	048	053	M	1.0
I	57	8	089	812	190	622	685	A	5.5 (CR), 1.5 (ED)
J	83	7	126	777	141	636	700	A	5.0

the AR C, there are three SAs, namely financial (FC), retail (RL), and waterfront (WF). Similarly, there are two SAs in AR D and AR I, namely core (CR) and edge (ED). The BP varies with respect to the SA of a corresponding AR and the TD (i.e., M, A, and E), as shown in [29]. We considered n as 10% higher than RS to create a more detailed and rigorous evaluation. We assumed a standard vehicle system for simulating the parking requests under smart on-street parking management. This system also assumes standard PSs and vehicle characteristics, namely types (i.e., electric vehicle, commercial vehicle, or oversized vehicle), length, width, and duration in time units. However, these are not explicitly shown in the simulation results.

We simulate the LUF algorithm using ten ARs of the Seattle city dataset and compare it with the FCFS, RR, and HPF algorithms in terms of four performance metrics, as shown in Table V. Note that there is no direct comparison between the proposed algorithm, LUF, and the existing literature. Therefore, we compare it with three baseline algorithms: FCFS, RR, and HPF. The simulation results indicate that the LUF algorithm consistently achieves superior performance compared to all other baseline algorithms.

We found that the LUF algorithm accepts 87.18%, 77.09%, 90.33%, 86.29%, 90.91%, 87.37%, 90.91%, 90.57%, 90.81% and 90.86% of parking requests on ten ARs, respectively. Moreover, the LUF algorithm improves the R_{acc} by 09.68%, 17.53%, 09.15%, 10.67%, 04.35%, 04.71%, and 13.37%,

TABLE V: Comparison of four performance metrics for FCFS, RR, HPF, and LUF algorithms

Algorithm	Parameter	A	B	C	D	E	F	G	H	I	J
FCFS	R_{acc}	031	037	405	317	060	075	050	046	594	561
	R_{rej}	008	011	122	084	006	020	005	007	091	139
	AV	10.03	04.51	02.68	10.27	04.02	03.70	03.66	01.86	07.66	08.79
	RV	0586.93	**0421.09**	2659.41	5807.61	0428.70	0533.40	0287.96	0141.15	8197.48	9100.94
RR	R_{acc}	031	037	405	317	060	075	050	046	594	561
	R_{rej}	008	011	122	084	006	020	005	007	091	139
	AV	10.03	05.85	03.49	10.27	04.02	03.70	03.66	01.86	07.66	08.79
	RV	0586.64	0420.88	2659.64	5807.70	0428.76	0532.93	0288.02	0141.20	8196.04	9101.48
HPF	R_{acc}	031	036	385	305	060	071	050	041	562	527
	R_{rej}	008	012	142	096	006	024	005	012	123	173
	AV	10.41	06.34	02.79	10.23	04.02	03.58	03.66	01.89	07.04	08.90
	RV	0484.34	0390.45	1889.35	5359.42	0428.75	0482.75	0272.05	0131.74	6962.98	8287.95
LUF	R_{acc}	034	037	476	346	060	083	050	048	622	636
	R_{rej}	005	011	051	055	006	12	005	005	063	064
	AV	10.04	05.85	03.53	10.53	04.02	3.50	3.66	01.94	07.73	08.91
	RV	**00588.27**	**00421.09**	**02929.31**	**06264.11**	**00430.27**	**00544.82**	**00288.80**	**00152.10**	**09116.59**	**10521.20**

compared to FCFS for AR A, C, D, F, H, I, and J, respectively. Similarly, it achieves improvements of 09.68%, 17.53%, 09.15%, 10.67%, 04.35%, 04.71%, and 13.37%, compared to RR for the same ARs. However, there is no improvement for ARs B, E, and G when compared to both FCFS and RR. Furthermore, the LUF algorithm enhances the R_{acc} by 09.68%, 02.78%, 23.64%, 13.44%, 16.90%, 17.07%, 10.68%, and 20.68% for AR A, B, C, D, F, H, I, and J, respectively, while no improvement is seen for AR E and G, compared to HPF. On the other hand, the LUF algorithm improves the RV by 00.23%, 10.16%, 07.87%, 00.37%, 02.14%, 00.29%, 07.75%, 11.22%, and 15.61%, compared to FCFS for ARs A, C, D, E, F, G, H, I, and J. However, no improvement is observed for AR B. When compared to RR, the improvements of the LUF algorithm are 00.28%, 00.05%, 10.16%, 07.86%, 00.35%, 02.24%, 00.27%, 07.71%, 11.26%, and 15.61% for ARs A, B, C, D, E, F, G, H, I, and J. Similarly, when compared to HPF, the LUF algorithm shows enhancements of 21.47%, 07.83%, 55.04%, 16.88%, 00.35%, 12.86%, 06.19%, 15.44%, 30.88%, and 26.97% for the respective ARs. Note that the best-performing results are shown in bold in Table V.

The rationality behind the better performance of the LUF algorithm is as follows. (1) Unlike the FCFS algorithm, the LUF algorithm considers the QoS of the parking requests. It increases the R_{acc} and RV. (2) Unlike the RR algorithm, the LUF algorithm selects the least utilized BF for assigning parking requests. It ensures that the PSs under the BFs are utilized efficiently, avoiding traffic congestion. (3) Unlike the HPF algorithm, the LUF algorithm selects the PS with the least FP for the parking requests without QoS, increasing user satisfaction. Overall, the LUF algorithm outperforms all ARs by optimizing parking request allocation and utilizing the PSs efficiently. Alternatively, the LUF algorithm improves the UZ of PSs in an equitable and equal manner. Moreover, it excels in minimizing R_{rej} and maximizing RV while maintaining user satisfaction, demonstrating its strength as smart on-street parking management in urban cities.

VI. Conclusion

We have introduced the QoS-aware LUF algorithm for smart on-street parking management. The algorithm aims to maximize the R_{acc} and RV and minimize the R_{rej} and AV. It handles parking requests with QoS and without QoS. It maps the parking requests to PSs by determining the least utilized BFs. We have used a real-time dataset of Seattle city with ten ARs and occupancy data to evaluate the performance of the LUF algorithm by conducting a series of simulation runs. We have compared the simulation results of the LUF algorithm with three baseline algorithms, FCFS, RR, and HPF, in terms of four performance metrics. We found that the proposed algorithm excels in all the performance metrics compared to the baseline algorithms. Specifically, the LUF algorithm shows an average improvement of 06.95% in R_{acc}, a 28.35% reduction in R_{rej}, a 06.52% decrease in AV, and a 05.56% increase in RV, compared to the FCFS algorithm. It also improves by 06.95% in R_{acc}, reduces R_{rej} by 28.35%, decreases AV by 00.50%, and increases RV by 05.57%, compared to the RR algorithm. Furthermore, compared to the HPF algorithm, it improves 11.49% in R_{acc}, reduces R_{rej} by 37.27%, decreases AV by 02.85%, and increases RV by 19.39%, irrespective of the ARs of the dataset, compared to the HPF algorithm.

The LUF algorithm can effectively balance performance with user satisfaction by giving equal importance to RV generation and fair use of PSs. Its ability to redirect traffic to less congested BFs while adapting to demand fluctuations makes it a better algorithm for real-time smart on-street parking management. However, the proposed algorithm utilizes simulated data that provides valuable insights to evaluate the performance of different algorithms; their real-world applicability is limited and can be modeled through machine learning techniques. Moreover, the proposed algorithm can be extended by considering real-time traffic and user behavior patterns. On the other hand, the proposed algorithm may occasionally map the few parking requests to slightly farther PSs in case they tend to full occupancy, which can increase user walking time or reduce user satisfaction.

References

[1] Meshari Aljohani, Stephan Olariu, Abrar Alali, and Shubham Jain. A survey of parking solutions for smart cities. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):10012–10029, 2021.

[2] Yanfeng Geng and Christos G Cassandras. New "smart parking" system based on resource allocation and reservations. *IEEE Transactions on intelligent transportation systems*, 14(3):1129–1139, 2013.

[3] Shivani Zoting. Smart parking systems market size, share, and trends 2025 to 2034. https://www.precedenceresearch.com/smart-parking-systems-market, 2025. [Online; accessed 18-April-2025].

[4] Sandeep Saharan, Seema Bawa, and Neeraj Kumar. Dynamic pricing techniques for intelligent transportation system in smart cities: A systematic review. *Computer Communications*, 150:603–625, 2020.

[5] Md Asif Thanedar and Sanjaya Kumar Panda. A dynamic resource management algorithm for maximizing service capability in fog-empowered vehicular ad-hoc networks. *Peer-to-Peer Networking and Applications*, 16(2):932–946, 2023.

[6] Md Asif Thanedar and Sanjaya Kumar Panda. Energy and priority-aware scheduling algorithm for handling delay-sensitive tasks in fog-enabled vehicular networks. *The Journal of Supercomputing*, 80(10):14346–14368, 2024.

[7] Md Asif Thanedar and Sanjaya Kumar Panda. An efficient resource orchestration algorithm for enhancing throughput in fog computing-enabled vehicular networks. *Vehicular Communications*, 53:100911, 2025.

[8] Md Asif Thanedar and Sanjaya Kumar Panda. An energy-efficient resource allocation algorithm for managing on-demand services in fog-enabled vehicular ad hoc networks. *International Journal of Web and Grid Services*, 20(2):135–158, 2024.

[9] Sohan Kumar Pande, Sanjaya Kumar Panda, Satyabrata Das, Mamoun Alazab, Kshira Sagar Sahoo, Ashish Kumar Luhach, and Anand Nayyar. A smart cloud service management algorithm for vehicular clouds. *IEEE Transactions on Intelligent Transportation Systems*, 22(8):5329–5340, 2020.

[10] Sourav Kumar Bhoi, Sanjaya Kumar Panda, Chittaranjan Mallick, and Kalyan Kumar Jena. Impact of road architecture and design on performance of city-based vanets. *Open Computer Science*, 11(1):423–436, 2021.

[11] Sandeep Saharan, Neeraj Kumar, and Seema Bawa. Dypark: A dynamic pricing and allocation scheme for smart on-street parking system. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):4217–4234, 2023.

[12] Sandeep Saharan, Seema Bawa, Neeraj Kumar, and Rajkumar Buyya. Foggy-park: A dynamic pricing and nsga based allocation scheme for on-street parking system. In *Proceedings of the SIGCOMM Workshop on Zero Trust Architecture for Next Generation Communications*, pages 31–36, 2024.

[13] Yi Zhang, Chih-Yu Wang, and Hung-Yu Wei. Parking reservation auction for parked vehicle assistance in vehicular fog computing. *IEEE transactions on vehicular technology*, 68(4):3126–3139, 2019.

[14] Amir O Kotb, Yao-Chun Shen, Xu Zhu, and Yi Huang. iparker—a new smart car-parking system based on dynamic resource allocation and pricing. *IEEE transactions on intelligent transportation systems*, 17(9):2637–2647, 2016.

[15] Vikas Hassija, Vikas Saxena, Vinay Chamola, and F Richard Yu. A parking slot allocation framework based on virtual voting and adaptive pricing algorithm. *IEEE Transactions on Vehicular Technology*, 69(6):5945–5957, 2020.

[16] Sourav Kumar Bhoi, Sanjaya Kumar Panda, and Debashee Tarai. Enhancing cpu performance using subcontrary mean dynamic round robin (smdrr) scheduling algorithm. *Journal of Global Research in Computer Science*, 2(12):17–21, 2011.

[17] Sanjaya Kumar Panda and Shidhanta Sen. Srra: a novel skewness-based algorithm for cloudlet scheduling. In *2023 IEEE 23rd international conference on software quality, reliability, and security (QRS)*, pages 772–781. IEEE, 2023.

[18] Sanjaya Kumar Panda, Debasis Dash, and Jitendra Kumar Rout. A group based time quantum round robin algorithm using min-max spread measure. *International Journal of Computer Applications*, 975:8887, 2013.

[19] Sanjaya Kumar Panda and Sourav Kumar Bhoi. An effective round robin algorithm using min-max dispersion measure. *International Journal on Computer Science and Engineering*, 4(1):45, 2012.

[20] Yanzhe Lei and Stefanus Jasin. Real-time dynamic pricing for revenue management with reusable resources and deterministic service time requirements. *University of Michigan*, 2016.

[21] Sabbir Ahmed, Mohammad Saidur Rahman, Mohammad Saiedur Rahaman, et al. A blockchain-based architecture for integrated smart parking systems. In *2019 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, pages 177–182. IEEE, 2019.

[22] Ho-Yin Mak. Enabling smarter cities with operations management. *Manufacturing & Service Operations Management*, 24(1):24–39, 2022.

[23] Munish Bhatia, Sandeep K Sood, and Simranpreet Kaur. Quantum-based predictive fog scheduler for iot applications. *Computers in Industry*, 111:51–67, 2019.

[24] Huanmei Qin, Ning Xu, Yonghuan Zhang, Qianqian Pang, and Zhaolin Lu. Research on parking recommendation methods considering travelers' decision behaviors and psychological characteristics. *Sustainability*, 15(8):6808, 2023.

[25] Zhenyu Mei, Zuchen Que, Hai Qiu, Zheng Zhu, and Zhengyi Cai. Optimizing the configuration of electric vehicle charging piles in public parking lots based on a multi-agent model. *Physica A: Statistical Mechanics and its Applications*, 632:129329, 2023.

[26] Ricard Garra, Santi Martínez, and Francesc Sebé. A privacy-preserving pay-by-phone parking system. *IEEE Transactions on vehicular technology*, 66(7):5697–5706, 2016.

[27] Dou An, Qingyu Yang, Donghe Li, Wei Yu, Wei Zhao, and Chao-Bo Yan. Where am i parking: Incentive online parking-space sharing mechanism with privacy protection. *IEEE Transactions on Automation Science and Engineering*, 19(1):143–162, 2020.

[28] Seattle Department of Transportation. Paid parking occupancy last 30 days. https://data.seattle.gov/Transportation/Paid-Parking-Occupancy-Last-30-Days-/rke9-rsvs/about_data, 2024. Accessed: Apr. 9, 2024.

[29] Seattle Department of Transportation. Seattle street parking rates. https://www.seattle.gov/transportation/projects-and-programs/programs/parking-program/paid-parking-information/street-parking-rates, 2024. Accessed: Apr. 9, 2024.