TRIMP: Three-Sided Stable Matching for Distributed Vehicle Sharing System using Stackelberg Game

Yang Xu, *Senior Member, IEEE,* Shanshan Zhang, *Graduate Student Member, IEEE,* Chen Lyu, *Member, IEEE,* Jia Liu, *Senior Member, IEEE,* Tarik Taleb, *Senior Member, IEEE,* and Shiratori Norio, *Life Fellow, IEEE*

Abstract—Distributed Vehicle Sharing System (DVSS) leverages emerging technologies such as blockchain to create a secure, transparent, and efficient platform for sharing vehicles. In such a system, both efficient matching of users with available vehicles and optimal pricing mechanisms play crucial roles in maximizing system revenue. However, most existing schemes utilize user-to-vehicle (two-sided) matching and pricing, which are unrealistic for DVSS due to the lack of participation of service providers. To address this issue, we propose in this paper a novel Three-sided stable Matching with an optimal Pricing (TRIMP) scheme. First, to achieve maximum utilities for all three parties simultaneously, we formulate the optimal policy and pricing problem as a three-stage Stackelberg game and derive its equilibrium points accordingly. Second, relying on these solutions from the Stackelberg game, we construct a three-sided cyclic matching for DVSS. Third, as the existence of such a matching is NP-complete, we design a specific vehicle sharing algorithm to realize stable matching. Extensive experiments demonstrate the effectiveness of our TRIMP scheme, which optimizes the matching process and ensures efficient resource allocation, leading to a more stable and well-functioning decentralized vehicle sharing ecosystem.

Index Terms—Distributed vehicle sharing, pricing, three-sided stable matching, Stackelberg game.

1 INTRODUCTION

WITH the rapid development of blockchain and other distributed technologies, the Distributed Vehicle Sharing System (DVSS) becomes a cutting-edge innovation aimed at revolutionizing vehicle sharing [1], [2]. By leveraging blockchain's inherent features, DVSS could ensure a secure, transparent, and efficient platform, which facilitates vehicle sharing without solely relying on one central authority. Within this platform, users, vehicle owners, and multiple service providers (SPs) interact directly, potentially using smart contracts [3], [4] to automate processes. For instance, a user could seamlessly book a vehicle provided by an SP through the DVSS platform, and the smart contract would automatically handle the reservation, payment, and even aspects like insurance verification, ensuring a hassle-free experience for all parties involved.

To realize such an appealing service, active user engagement and achievement of revenue optimization are imperative [5]–[7]. In this context, both efficient user-vehicle matching and implementation of optimal pricing mechanisms play pivotal roles. On the one hand, efficient user-vehicle matching facilitates the rapid pairing of users with available vehicles. This process takes into account variables (e.g., user location, vehicle availability, and user preferences [8]), ensuring seamless connections. On the other hand, the establishment of optimal pricing mechanisms entails the utilization of dynamic pricing models that adapt to real-time demand, traffic conditions, and user urgency. This strategy strikes a delicate balance between user affordability and system profitability. Therefore, by delivering impeccable user experiences through precise matching and strategic pricing adjustments, DVSS can simultaneously enhance user attraction and augment revenue generation, leading to a sustainable operational system.

Today's vehicle sharing system solely makes use of traditional two-sided approaches to determine the user-vehicle pricing [9]–[16] and matching [17]–[23], involving two groups of participants, i.e., users and vehicle owners. However, such approaches are limited to centralized vehicle sharing systems. In the DVSS, we have to involve SPs in the new three-sided market since it has three distinct groups of participants. However, it introduces several challenging issues.

First, the pricing problem among SPs, vehicle owners, and users transforms into a dynamic sequential decision-making process [24], [25] for three parties. Each of the three parties engages

- *Y. Xu and S. Zhang are with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: yxu@xidian.edu.cn; zhangshanshan@stu.xidian.edu.cn).*
- *C. Lyu is with the MoE Key Laboratory of Interdisciplinary Research of Computation and Economics, Shanghai University of Finance and Economics, Shanghai 200433, China (email: lyu.chen@mail.shufe.edu.cn).*
- *J. Liu is with the Center for Strategic Cyber Resilience Research and Development, National Institute of Informatics, Tokyo 101-8430, Japan (e-mail: jliu@nii.ac.jp).*
- *T. Taleb is with Ruhr University Bochum, Bochum, 44801 Germany (e-mail: tarik.taleb@rub.de).*
- *N. Shiratori is with the Research and Development Initiative, Chuo University, Tokyo 112-8551, Japan (e-mail: norio.shiratori.e8@tohoku.ac.jp).*

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62372361; in part by JSPS KAKENHI Grant Number JP23K16877; in part by the Project of Cyber Security Establishment with Inter-University Cooperation; in part by the Fundamental Research Funds for the Central Universities under Grant No. 2023110139; in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY24F020011; and in part by the European Union's under the 6G-Path project (Grant No. 101139172) and by the AerOS project (Grant No. 101069732). This work was partially conducted in ICTFICIAL Oy. The paper reflects only the authors' views, and the European Commission bears no responsibility for any utilization of the information contained herein.
(Corresponding authors: Chen Lyu; Jia Liu)

in the sequential decision-making process, where the decisions of one party significantly impact the options and preferences of the others. Moreover, in the context of such a dynamic process, it necessitates achieving maximum utilities for all three parties simultaneously in order to optimize user attraction and revenue generation for the DVSS.

Second, the three-sided matching mechanism collaborates with pricing mechanisms, posing a challenge often classified as NP-hard to solve [26]–[28]. Traditional exhaustive search methods, which explore all possible combinations, become impractical as the complexity grows [29], [30]. These methods often fail to ensure timely solutions due to their intricate nature. Therefore, an approximate algorithm is needed to provide effective and timely solutions for addressing this problem.

Third, emphasizing the stability of three-sided matching becomes paramount as we aim to guarantee robust and reliable outcomes in the dynamic and distributed environment of the DVSS. Achieving stability is particularly challenging due to the potential presence of incomplete information about participants in the vehicle sharing system. In such an environment, uncertainties related to user preferences, vehicle availability, or service provider strategies may prevail. Designing a matching algorithm that can operate effectively under these conditions is crucial. Traditional stability concepts, such as the stability of marriages in the classic stable marriage problem [31], cannot be directly used for the unique dynamics and uncertainties of the DVSS. Therefore, it is imperative to develop a new stable algorithm that accounts for incomplete information, ensuring the reliability and long-term stability of the three-sided matching process in the DVSS.

In this paper, we propose a Three-sided stable Matching with an optimal Pricing (TRIMP) scheme, offering a comprehensive solution to all the above challenges posed by the DVSS. Our system takes into account interactive pricing decisions among three participants in the DVSS, ensuring stability and efficiency in the final three-sided matching outcomes. TRIMP is designed to optimize user attraction, maximize revenue generation, and navigate the dynamic sequential decision-making process inherent in the DVSS. First, we propose a pricing mechanism based on the three-stage Stackelberg game [24], [32] to derive an optimal pricing strategy within the DVSS. Our mechanism effectively simulates dynamic sequential decision-making processes of vehicle owners, SPs, and users, ultimately leading to the establishment of a game equilibrium. Both theoretical proof and numerical simulations are provided to prove the existence and uniqueness of the Stackelberg equilibrium. Second, building upon the optimal pricing mechanism, we formulate a matching model based on a three-sided cyclic matching game [27] designed for SPs, vehicle owners, and users within the DVSS. This model is meticulously crafted to accommodate the interactions among three participants, providing a comprehensive framework for pricing and matching in the dynamics of the DVSS. Third, recognizing the inherent NP-hardness of the matching model, we design a novel three-sided matching algorithm capable of approximately achieving optimal stable matching results. The algorithm's stability is rigorously substantiated through theoretical proofs. Especially, our algorithm operates based on participants' preference lists, offering a practical advantage by alleviating the requirement for complete information about all participants in the DVSS. This strategic approach enhances the algorithm's adaptability to real-world scenarios where information may be incomplete. Finally, through extensive comparative evaluations, we demonstrate that TRIMP consistently outperforms four alternative matching schemes. It not only attains stability but also maximizes total utilities for participants within the DVSS, thereby highlighting its effectiveness and stability in dynamic and distributed environments.

To the best of our knowledge, this is the first stable matching for distributed vehicle sharing services. The main contributions are summarized as follows:

- We propose TRIMP as a novel solution designed to effectively tackle challenges for the unique three-sided market in the DVSS environment. TRIMP comprises two essential components: an optimal pricing mechanism and a three-sided matching mechanism, addressing the complexities of pricing, matching, and stability in the dynamic and distributed environment of the DVSS.
- We formulate the pricing problem as a three-stage Stackelberg game. Employing a backward induction approach, we propose an optimal pricing mechanism to achieve game equilibrium. This equilibrium ensures that no participant can unilaterally improve its utility by deviating from its optimal pricing strategy.
- Building upon the optimal pricing mechanism, we model the matching problem as a three-sided cyclic matching game for the DVSS. To address this NP-hard problem, we design a three-sided matching mechanism and provide theoretical proof of its stability. Especially, our algorithm solely relies on participants' preference lists, eliminating the need for complete information about all participants in the DVSS.
- To comprehensively evaluate TRIMP's performance, we employ numerical simulations to illustrate the dynamic adjustment process within the Stackelberg pricing game. These simulations also highlight the achieved optimal stable matching outcomes by TRIMP, validating its effectiveness, scalability, and efficiency for distributed vehicle sharing services.

The rest of the paper is organized as follows. Section 2 provides an overview of related work. In Section 3, we introduce the system model and the blockchain-based vehicle sharing process. We propose the game modeling and problem formulation in Section 4. The implementation details of our TRIMP scheme are presented in Section 5, with experimental results discussed in Section 6. At last, we conclude our work in Section 7.

2 Related Work

Extensive research has been dedicated to the realm of vehicle sharing, with primary areas of research encompassing vehicle transaction pricing, optimal user-to-vehicle matching, resource allocation and scheduling, and the enhancement of security in the DVSS.

2.1 Pricing

Considering users are usually price sensitive, pricing strategy design is an important component of the vehicle sharing system and is closely related to vehicle sharing transactions. In the research of pricing strategies, the primary emphasis lies in maximizing the overall utility of the system [9]–[16]. Regarding the charging and pricing strategies within the domain of shared electric vehicles, Xie *et al.* [9] formulated the problem of the overall decision of platform SP to optimize the SP's total profit. Ren *et al.* [10] accounted for users' price sensitivity and formulated a dynamic pricing strategy aimed at maximizing the total profit of SPs. Kamatani *et al.* [11] proposed an approach that employs reinforcement learning to devise a dynamic pricing scheme. Their strategy aims to tackle issues arising from unpredictable shifts in user demand and imbalanced vehicle distribution, ultimately optimizing vehicle utilization and boosting the overall revenue of SP. Banerjee *et al.* [12] presented an approximation framework tailored for vehicle sharing systems. Their approach provides a unified strategy for diverse controls, including pricing and matching, objective functions, and system constraints, employing natural convex relaxations. The resulting nonasymptotic and parametric guarantees offer practical insights for system design.

Wang *et al.* [13] proposed a pricing scheme based on demand prediction for shared electric vehicles in a large-scale vehicle sharing network. Their scheme identifies the optimal amalgamation of two tiers of price adjustments, strategically incentivizing the spatial and temporal distribution of vehicular traffic to achieve the maximization of system profitability. For shared vehicles, Pfrommer *et al.* [14] integrated real-time price incentives with route decision-making in shared mobility systems. Utilizing a predictive control approach, they periodically recalculated route directions to balance user payment incentives and vehicle allocation costs. Waserhole *et al.* [15] utilized pricing incentives to improve the efficiency of vehicle sharing systems. They formulated a Markov model that depicts a closed queuing network with constraints on buffer size and time-dependent service duration. Yang *et al.* [16] established a two-stage Stackelberg model, incorporating government involvement to ascertain both subsidy rates and pricing strategies. This model seeks to stimulate user engagement, safeguard the interests of governmental entities and operators, and ultimately attain an optimal level of overall efficiency.

The aforementioned studies primarily focused on two-sided pricing models for vehicle sharing, a framework that may not be applicable to the DVSS due to its unique three-participant structure. As a result, there is a need for a specialized approach that addresses the dynamics and challenges introduced by the involvement of three key participants in the DVSS.

2.2 Matching

Another crucial aspect lies in the matching process within the vehicle sharing system. Optimizing the matching algorithm becomes essential for ensuring efficient interactions among system participants. Peng *et al.* [17] incorporated a payment system that addresses both equity and incentives and then proposed a matching model in ride-sharing to minimize commuters' costs. Chau *et al.* [18] designed a decentralized vehicle sharing matching mechanism based on fair cost sharing, which induces the optimal stable matching result and can achieve social optimal with minimum total cost. Zhang *et al.* [19] considered the preferences of customers towards peers in vehicle sharing and established a matching strategy based on user preferences, which can improve the individual efficiency of users while maximizing overall efficiency.

Pelzer *et al.* [20] presented a road network zoning-based method for customer matching in a large and dynamic vehicle pooling system, in order to maximize pooling potential while managing detours within specified limits. Yatnalka *et al.* [21] developed a matching model for vehicle sharing, which incorporates a user threshold time to pick potential matches and then optimizes the matching process based on user features. Wang *et al.* [22] formulated the matching problem for a dynamic vehicle sharing system between drivers and customers, addressing the challenge of incomplete information by constructing a dynamic stable matching for vehicle sharing. Rasulkhani *et al.* [23] put forward a matching model between user groups and route sets in a network, incorporating market equilibrium pricing and yielding stable outcomes for user groups.

Nevertheless, these efforts primarily focused on pairing available idle vehicles with users in need or matching appropriate drivers with users requiring services, which may only be suitable for two-sided matching scenarios.

2.3 Resource Allocation and Security

Other research issues related to vehicle sharing include resource allocation scheduling [33]–[36] and security issues of the DVSS [1], [37]–[39].

Fanti *et al.* [33] explored the fleet size issue in an electric car sharing system by modeling it as a discrete event system within a queueing network, where vehicle utilization was considered and an optimization problem was formulated to maximize system revenue through optimal fleet sizing. Nourinejad *et al.* [34] focused on the alignment of tactical and operational decisions in fleet sizing for vehicle sharing. They proposed an efficient approach by integrating two integer programming models to plan the fleet size based on demand and schedule relocation operations. Wang *et al.* [36] tackled the rebalancing problem of one-way shared electric vehicles by introducing a zero-one nonlinear programming model. Their model aims to minimize the total cost while ensuring a minimum number of low-battery vehicles and unbalanced vehicles.

In striving for a balance between efficiency and security in the blockchain-enabled framework, Ning *et al.* [1] proposed an algorithm based on deep reinforcement learning. Their algorithm selects active miners and transactions, achieving a tradeoff between latency and blockchain security. Valaštín *et al.* [37] designed a decentralized peer-to-peer car-sharing application using blockchain and smart contracts, reducing system costs and improving data transparency in DVSS. Kim *et al.* [38] introduced a secure authentication scheme for DVSS. Their scheme withstands diverse attacks such as impersonation and replay attacks, ensuring both secure mutual authentication and privacy preservation.

3 System Model and Blockchain-based Vehicle Sharing Process

3.1 System Model

The Decentralized Vehicle Sharing System (DVSS) is a revolutionary platform designed to facilitate city vehicle sharing services, bringing together three distinct groups of participants: renters (or users), vehicle owners, and SPs. This innovative system leverages blockchain technology and smart contracts to streamline the process of vehicle reservations and ensure secure, transparent transactions.

- **Renters (or Users)**: These are individuals seeking short-term vehicle rentals for their transportation needs within the city. $\mathcal{R} = \{1, 2, \cdots, r, \cdots, R\}$ indicates the set of renters, with $r \in \mathcal{R}$ being a unique renter identifier.
- **Vehicle Owners**: Vehicle owners are individuals who make their vehicles available for rent through the DVSS platform when they are not using them. We denote the set of vehicles available for rent as $\mathcal{V} = \{1, 2, \cdots, v, \cdots, V\}$, where $v \in \mathcal{V}$ represents a unique vehicle identifier. This notation

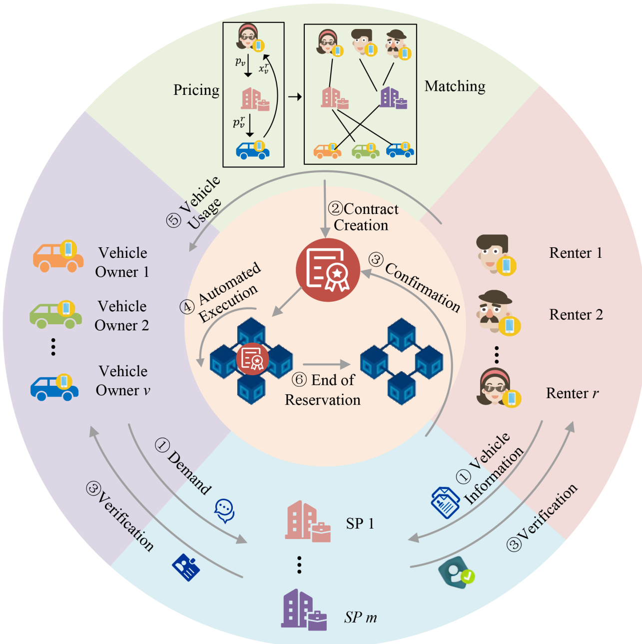


Fig. 1. System model and blockchain-based vehicle sharing process.

accommodates situations where “Vehicle Owners” and “Vehicles” may be used interchangeably in the paper.

- **SPs**: The SPs or platform companies play a vital role in the DVSS ecosystem by acting as intermediaries that facilitate the leasing transactions between renters and vehicle owners. They ensure a smooth process and guarantee the transactions’ legitimacy. The set of platform companies is given by $\mathcal{M} = \{1, 2, \cdots, m, \cdots, M\}$, where $m \in \mathcal{M}$ represents a unique SP identifier.

3.2 Blockchain-based Vehicle Sharing Process

As shown in Fig. 1, we propose the vehicle sharing process using blockchain’s smart contracts in the DVSS. The detailed steps are as follows:

1) Demand and Vehicle Information: Renters may run a mobile application to express their demand for vehicle rentals, specifying their preferences for the type of vehicle, duration, and location. Vehicle owners could also run the application to provide information about their available vehicles, including their availability schedules and rental terms.
2) Smart Contract Creation: Upon implementing the matching and pricing mechanisms, a renter will discover a suitable vehicle along with the corresponding platform SP. A smart contract is then created on the blockchain. This smart contract contains the terms of the rental agreement, including the rental duration, payment details, and conditions.
3) Verification and Confirmation: The SP verifies the renter’s information and checks the availability of the chosen vehicle. Once verified, the SP confirms the reservation by interacting with the smart contract.
4) Automated Execution: Upon the reservation’s start time, the smart contract automatically executes. It locks the agreed-upon payment amount in escrow and grants the renter access to the vehicle using a secure digital key.

TABLE 1
Key Notations and Descriptions.

Notation	Description
$\mathcal{R}, \mathcal{M}, \mathcal{V}$	The set of renters, SPs, and vehicles (w/wo owner)
r, v, m	The specific renter $r \in \mathcal{R}$, SP $m \in \mathcal{M}$, vehicle (w/wo owner) $v \in \mathcal{V}$
c_v	The cost of the vehicle owner for sharing vehicle v
p_v	The reward of the vehicle owner for sharing the vehicle v
x_v^r	The willingness of renter r to lease vehicle v
p_v^r	The price for renter r to lease the vehicle v
$M(p_v)$	The cost of operating a vehicle sharing service
a, b	The operating cost parameters
$S(x_v^r)$	The satisfaction of renter r to the vehicle v
$Q(d_v, m_v)$	The vehicle v’s basic situation
d_v,	The driving mileage of vehicle v
m_v	The usage time of vehicle v
α	The renter satisfaction coefficient
β_1, β_2	The weight coefficients of driving mileage and usage time, respectively
U_v, U_m, U_r	The utility of vehicle owner v, SP m, and renter r
$(p_v^*, (p_v^r)^*, (x_v^r)^*)$	The optimal reward strategy, price strategy, and request strategy group
$Z_{m,v}$	Binary variable indicating whether SP m chooses to serve vehicle v
$Y_{v,r}$	Binary variable indicating whether vehicle v is rented to renter r
t_r	The rental time for the renter r
t_v	The idle time for the vehicle v
N_m	The maximum number of vehicles that can be serviced by SP m
$\mathcal{D}(m), \mathcal{D}(v), \mathcal{D}(r)$	The matched vehicles of SP m, the matched renter of vehicle v, and the matched SP of renter r
$N(\mathcal{G}, \cdot)$	The number of partners matched by SP m, vehicle v, or renter r
PL_m, PL_v, PL_r	The preference list of SP m, vehicle v, and renter v
$A^{+1}(\mathcal{G}, m)$	The set of all vehicles preferred by SP m over its current matching $\mathcal{G}(m)$
$A^{+1}(\mathcal{G}, v)$	The set of all renters preferred by vehicle v over its current matching $\mathcal{G}(v)$
$A^{-1}(\mathcal{G}, m)$	The set of all renters can be served by SP m
$A^{-2}(\mathcal{G}, m)$	The set of all vehicles for which there exists a renter r that vehicle v prefers over its current partner $\mathcal{G}(v)$, and renter r can still be served by SP m

5) Vehicle Usage: The renter can access and use the reserved vehicle during the agreed-upon rental period. The vehicle’s usage is tracked through the smart contract.
6) End of Reservation: At the end of the rental period, the smart contract releases the escrowed payment to the vehicle owner and revokes the renter’s access to the vehicle.

By employing blockchain’s smart contracts, DVSS could ensure that reservations are secure, transparent, and tamper-proof. Renters, vehicle owners, and SPs can engage in transactions confidently, knowing that the terms of the rental agreement are enforced automatically. Therefore, this model could foster a more efficient and trustworthy ecosystem for all participants.

Design Goal: In the above system, we concentrate on the design of pricing and matching mechanisms in the three-sided market. First, we need to implement the pricing mechanisms among renters, vehicle owners, and SPs. After determining the pricing strategies, we need to achieve three-sided matching for all the participants.

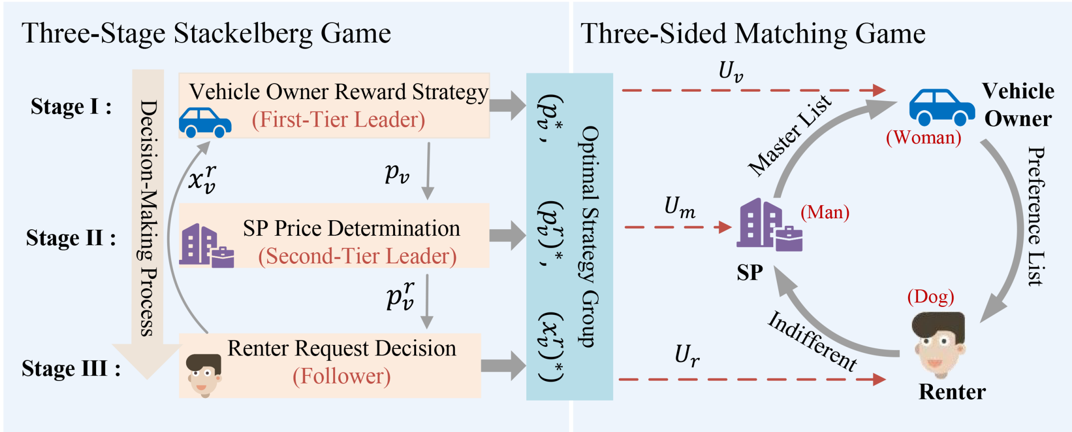


Fig. 2. Game modeling for dynamic interactions among participants in the DVSS.

4 Game Modeling and Problem Formulation

4.1 Determination of Utility Functions

As illustrated in Fig. 2, we model the dynamic interactions among renters, vehicle owners, and SPs in the DVSS framework as a combination of a three-stage Stackelberg game (for pricing) and a three-sided matching game (for matching). The key notations we use in this paper are presented in Table 1.

For vehicle owners, once vehicle v has been rented out for a day, we assume the cost of vehicle sharing, such as depreciation, to be c_v, and the reward of vehicle sharing to be p_v. Consequently, vehicle owners are inclined to offer their vehicles for rent only when $p_v > c_v$. Let $x_v^r \in [0, 1]$ denote the willingness of renter r to lease vehicle v. Then, the utility function of vehicle v for renting to renter r can be defined as

$$U_v(p_v, x_v^r) = (p_v - c_v)x_v^r. \tag{1}$$

Once a lease transaction is confirmed, an SP will charge fees for vehicle sharing service, resulting in the renter r paying a higher price p_v^r to rent vehicle v. Let $M(p_v)$ denote the cost of operating a vehicle sharing service, which we consider typically consists of both fixed and variable components. The fixed cost generally encompasses labor and operational expenses, which we represent with the parameter b. The variable cost includes depreciation, insurance, and maintenance, and it is consistently proportional to the rental reward p_v with coefficient a. Then, $M(p_v)$ is expressed as

$$M(p_v) = a \cdot p_v + b. \tag{2}$$

Regarding the determination of parameters a and b, we can collect historical data on vehicle usage, maintenance cost, fuel consumption, and other operational expenses, and then perform regression analysis. Parameters a and b are interpreted as coefficients in a linear regression model where the dependent variable is $M(p_v)$ and the independent variable is p_v. We can further validate the estimates of a and b by comparing the predicted cost with the actual observed cost, and make appropriate calibrations to improve the accuracy of the estimates.

Based on $M(p_v)$, the utility function of SP could be defined as follows:

$$U_m(p_v, p_v^r, x_v^r) = \left[\left(p_v^r - M(p_v) - p_v\right) x_v^r\right]^+, \tag{3}$$

where $[*]^+ = \max\{0, *\}$.

We use the satisfaction $S(x_v^r)$, to comprehensively evaluate the benefits that renter r can obtain from renting vehicle v. In order to capture more accurately the impacts of multiple factors on satisfaction, we express $S(x_v^r)$ as

$$S(x_v^r) = \alpha \ln(1 + x_v^r) Q(d_v, m_v). \tag{4}$$

Eq. (4) consists of two parts. The first term, $\alpha \ln(1 + x_v^r)$, reflects the relationship between satisfaction and willingness. This is a generally accepted satisfaction expression [40], where α represents the renter's satisfaction coefficient, and satisfaction is a concave function of the willingness to rent the vehicle, capturing the marginal diminishing effect of willingness on satisfaction. The second term, $Q(d_v, m_v) = \frac{\beta_1}{d_v} + \frac{\beta_2}{m_v}$, describes the basic condition of vehicle v, which is inversely proportional to its cumulative driving mileage d_v and usage time m_v. This is because higher mileage and longer usage time suggest older age or lower performance of the vehicle. The parameters β_1 and β_2 quantify the renters' sensitivity to the vehicle's driving mileage and usage time, respectively. Larger values of β_1 and β_2 indicate greater concern for the vehicle's condition, resulting in a larger impact on satisfaction.

Note that in Eq. (4), the basic condition of a vehicle is modeled depending on two parameters d_v (driving mileage) and m_v (usage time). We consider these parameters as they are intuitive indicators of vehicle wear and aging, and thus are critical factors in determining the vehicle condition. A more comprehensive model can integrate more factors, such as brand reputation, maintenance history, recent repairs, external conditions, etc., to more accurately and comprehensively evaluate the impact of vehicle condition on renter's satisfaction. However, incorporating more parameters to model the basic condition of the vehicle will not affect the overall architecture of TRIMP in this work.

According to $S(x_v^r)$, the utility function of renter r is given by:

$$U_r(p_v^r, x_v^r) = \left[S(x_v^r) - p_v^r x_v^r\right]^+. \tag{5}$$

4.2 Pricing Game Modeling

According to the utility functions defined in equations (1), (3), and (5), we model the pricing mechanism in the DVSS as a three-stage Stackelberg game and then determine the optimal strategy group $(p_v^*, (p_v^r)^*, (x_v^r)^*)$. This is because the three-stage Stackelberg game offers advantages by allowing a leader to make decisions before followers, enabling sequential decision-making. This structure leverages the leader's ability to strategically shape the followers' best responses, capitalizing on asymmetric information. Additionally, the model captures dynamic decision processes and provides insights into our scenarios where temporal sequencing is crucial.

In our model, a vehicle owner v can adjust the quoted price p_v to affect the profit of an SP, and the SP can set a higher renting price p_v^r to maximize its utility. Meanwhile, a renter's leasing intention, based on the renting price p_v^r, influences the utility of the vehicle owner and SP. Therefore, in the initial stage, the vehicle owner v acts as the first-tier leader, strategically determining the optimal reward strategy p_v^* to maximize its utility $U_v(p_v, x_v^r)$. Once each renter has established its optimal strategy, it ensures that the owner cannot enhance its utility by choosing any alternative strategy p_v. Similarly, the SP acts as the second-tier leader, deciding a pricing strategy $(p_v^r)^*$ to maximize its utility given the owner's strategy p_v^*. In the third stage, the renter becomes the follower and customizes an optimal strategy $(x_v^r)^*$ under the given pricing strategy $(p_v^r)^*$.

The above pricing and request process is devoted to finding an optimal strategy group $(p_v^*, (p_v^r)^*, (x_v^r)^*)$ guarantees that no participant can improve its utility by unilaterally deviating from its optimal strategy, i.e.,

Stage I (Vehicle Owner): $U_v(p_v^*, (x_v^r)^*) \geqslant U_v(p_v, (x_v^r)^*)$, (6)

Stage II (SP): $U_m(p_v^*, (p_v^r)^*, (x_v^r)^*) \geqslant U_m(p_v^*, p_v^r, (x_v^r)^*)$, (7)

Stage III (Renter): $U_r((p_v^r)^*, (x_v^r)^*) \geqslant U_r((p_v^r)^*, x_v^r)$. (8)

In the context of the proposed Stackelberg game, the three inequalities mentioned above signify the equilibrium conditions. We formally define Stackelberg Equilibrium (SE) as follows:

Definition 1 ***Stackelberg Equilibrium (SE)****. A strategy group $(p_v^*, (p_v^r)^*, (x_v^r)^*)$, where $p_v > c_v$, constitutes a Stackelberg equilibrium in the three-stage Stackelberg game if it satisfies inequalities (6), (7), and (8) simultaneously.*

4.3 Matching Problem Formulation

Building upon the optimal request strategies and pricing determined in the previous section, our objective is to maximize the total utility in the market. We specifically concentrate on the challenge of effectively matching vehicles, SPs, and renters in the DVSS. On the one hand, the matching between SPs (i.e., $\mathcal{M}$) and vehicles (i.e., $\mathcal{V}$) is characterized by a one-to-many matching, where each SP can be matched with multiple vehicles. On the other hand, the matching between vehicles (i.e., $\mathcal{V}$) and renters (i.e., $\mathcal{R}$) is a one-to-one matching, where each vehicle is matched with a single renter.

We begin by introducing some variables. Let binary variable $Z_{m,v}$ indicate whether SP m chooses to serve vehicle v, and let binary variable $Y_{v,r}$ indicate whether vehicle v is rented to renter r. The detailed definitions are as follows:

$$Z_{m,v} = \begin{cases} 1, & \text{if SP } m \text{ choose to serve vehicle } v, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

and

$$Y_{v,r} = \begin{cases} 1, & \text{if vehicle } v \text{ is rented to renter } r, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

To evaluate the market performance, we establish performance metrics from two perspectives: SP utility and vehicle revenue. The utility function of SP m from vehicle v can be re-expressed as

$$U_m(v,r) = \sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}} Z_{m,v}Y_{v,r}U_m(p_v^*, (p_v^r)^*, (x_v^r)^*). \tag{11}$$

Accordingly, we re-express the vehicle revenue from the renter as

$$U_v(m,r) = \sum_{r\in\mathcal{R}} Y_{v,r}U_v(p_v^*, (x_v^r)^*). \tag{12}$$

Then, we formulate the overall optimization problem in the considered DVSS as follows:

$$\max_{\mathbf{Z},\mathbf{Y}} \quad \sum_{v\in\mathcal{V}}\sum_{r\in\mathcal{R}}\sum_{m\in\mathcal{M}} \left(U_m(v,r) + U_v(m,r)\right) \tag{13a}$$
$$\text{s.t.} \quad Z_{m,v} \in \{0,1\}, \tag{13b}$$
$$Y_{v,r} \in \{0,1\}, \tag{13c}$$
$$t_r \leqslant t_v, \forall r \in \mathcal{R}, \forall v \in \mathcal{V}, \tag{13d}$$
$$\sum_{v\in\mathcal{V}} Z_{m,v} \leqslant N_m, \forall m \in \mathcal{M}, \tag{13e}$$
$$\sum_{r\in\mathcal{R}} Y_{v,r} \leqslant 1, \forall v \in \mathcal{V}, \tag{13f}$$

where constraint (13d) ensures that the rental time t_r for a renter r should be less than or equal to the idle time t_v of the vehicle v that is matched to the renter. N_m represents the maximum number of vehicles that can be serviced by SP m. Constraint (13e) indicates that the number of vehicles matched by the SP cannot exceed the maximum number of vehicles that can be serviced. Constraint (13f) indicates that each vehicle can only be rented to one renter at the same time, resulting in a one-to-one matching between vehicles and renters.

Remark 1 *Note that the pricing process is modeled as a three-stage Stackelberg game, which is a sequential game where players decide their best strategies in sequence. After the vehicle and SP set the reward and price respectively, the best strategy and optimal utility of the renter, who acts as the follower, will be determined. In other words, when optimizing the utilities of the vehicle and SP, the sequential decision-making process inherently incorporates the optimization of the renter's utility. Moreover, in the DVSS matching process that is elaborated upon in the next section, once the matching between the SP and the vehicle (i.e., two-sided) is completed, the matching among the SP, vehicle, and renter (i.e., three-sided) will also be determined. It implies that optimizing the utilities of the vehicle and SP naturally covers the optimization of the renter's utility. Therefore, we explicitly encompass only the utilities of the vehicle and SP in the formulation of problem (13).*

The optimization problem (13) in the DVSS is a Mixed Integer Non-Linear Programming (MINLP) problem, a class of problems generally known to be NP-hard to solve [26]–[28]. Finding a solution poses a considerable challenge, particularly in the context of large-scale vehicle sharing service networks. Exhaustive search methods are highly complex and often fail to guarantee a solution within an acceptable time frame [29], [30]. To this end, we develop a Three-sided Stable Matching with an Optimal Pricing (TRIMP) scheme, which will be elaborated upon in the next section.

5 Three-sided Stable Matching with an Optimal Pricing Scheme

In this section, we present the details of our TRIMP scheme, comprising an optimal pricing mechanism and a distributed three-sided stable matching mechanism based on the Three-Dimensional Stable Marriage model. We will introduce each of them sequentially.

5.1 Optimal Pricing Determination

To tackle the three-stage Stackelberg game, we deploy a backward induction approach. We briefly explain how it works in our three-stage Stackelberg game. The attainment of the SE needs to explore the sub-games within each layer. First, in Stage III, we determine the renter's optimal request strategy $(x_v^r)^*$ under a specified pricing strategy p_v^r. Second, proceeding to Stage II, we deduce the SP's optimal pricing strategy $(p_v^r)^*$ when provided with a reward strategy p_v. Finally, in Stage I, we investigate the vehicle owner's optimal reward strategy p_v^*. Additionally, it is crucial to prove that $(p_v^*, (p_v^r)^*, (x_v^r)^*)$ forms a unique SE, ensuring that no party has an incentive to unilaterally deviate from their optimal decisions.

5.1.1 Analytical Framework

First, let's delve into the analysis of Stage III, focusing on the perspective of the renter, to ascertain the optimal request strategy $(x_v^r)^*$. Lemma 1 provides a solution for this optimization process.

Lemma 1 *In Stage III, given any pricing strategy p_v^r, the optimal strategy of the renter r (i.e., the optimal request for the vehicle v) can be determined by*

$$(x_v^r)^* = \left[\frac{\alpha Q(d_v, m_v)}{p_v^r} - 1\right]^+. \tag{14}$$

Proof: We take the first-order and second-order derivatives of $U(p_v^r, x_v^r)$ with respect to x_v^r, yielding

$$\frac{\partial U_r(p_v^r, x_v^r)}{\partial x_v^r} = \frac{\alpha Q(d_v, m_v)}{1 + x_v^r} - p_v^r, \tag{15}$$

$$\frac{\partial^2 U_r(p_v^r, x_v^r)}{\partial (x_v^r)^2} = -\frac{\alpha Q(d_v, m_v)}{(1 + x_v^r)^2} < 0. \tag{16}$$

We can observe that $U_r(p_v^r, x_v^r)$ is continuous and differentiable on x_v^r. Moreover, the second-order derivative of $U_r(p_v^r, x_v^r)$ with respect to x_v^r is negative. Therefore, $U_r(p_v^r, x_v^r)$ is a concave function of x_v^r, satisfying Eq. (8). This also indicates that we can derive the optimal strategy $(x_v^r)^*$ for the renter by solving $\frac{\partial U_r(p_v^r, x_v^r)}{\partial x_v^r} = 0$. This completes the proof. □

Second, we analyze Stage II to unveil the optimal pricing strategy $(p_v^r)^*$ for the SP. Lemma 2 provides a solution for this strategic determination.

Lemma 2 *In Stage II, given any pricing strategy p_v, the optimal strategy $(p_v^r)^*$ of SP can be determined by solving $\frac{\partial U_m(p_v, p_v^r, (x_v^r)^*)}{\partial p_v^r} = 0$, which is expressed as*

$$(p_v^r)^* = \sqrt{\alpha Q(d_v, m_v)(p_v + M(p_v))}, \tag{17}$$

where $(p_v^r)^* \in (p_{vmin}^r, p_{vmax}^r)$, ensuring that $U_m(p_v, p_v^r, x_v^r)$ and $U_r(p_v^r, x_v^r)$ remain positive, and we have

$$p_{vmax}^r = \frac{\alpha \ln(1 + x_v^r) Q(d_v, m_v)}{x_v^r}, \tag{18}$$

$$p_{vmin}^r = M(p_v) + p_v. \tag{19}$$

Proof: By Substituting $(x_v^r)^*$ into Eq. (3), we have

$$U_m(p_v, p_v^r, (x_v^r)^*) = \left[(p_v^r - M(p_v) - p_v)\left(\frac{\alpha Q(d_v, m_v)}{p_v^r} - 1\right)\right]^+.$$

To find the SE, we derive the first-order derivative of $U_m(p_v, p_v^r, (x_v^r)^*)$ with respect to p_v^r, which yields

$$\frac{\partial U_m(p_v, p_v^r, (x_v^r)^*)}{\partial p_v^r} = \frac{\alpha Q(d_v, m_v)(M(p_v) + p_v)}{(p_v^r)^2} - 1.$$

Taking the second-order derivative of $U_m(p_v, p_v^r, (x_v^r)^*)$ with respect to p_v^r, we have

$$\begin{aligned}\frac{\partial^2 U_m(p_v, p_v^r, (x_v^r)^*)}{\partial (p_v^r)^2} &= -\frac{2\alpha Q(d_v, m_v)(p_v + M(p_v))}{(p_v^r)^3} \\ &< 0.\end{aligned} \tag{20}$$

We can conclude that $U_m(p_v, p_v^r, (x_v^r)^*)$ is a concave function of p_v^r, satisfying Eq. (7). Therefore, the optimal value $(p_v^r)^*$ can be obtained by solving $\frac{\partial U_m(p_v, p_v^r, (x_v^r)^*)}{\partial p_v^r} = 0$. This completes the proof. □

Finally, we study Stage I to determine the optimal reward strategy p_v^* of the vehicle owner. Lemma 3 provides how to obtain the optimal pricing strategy.

Lemma 3 *In Stage I, the optimal strategy p_v^* of the vehicle owner can be expressed as*

$$p_v^* = \arg\max_{p_v} U_v(p_v, (x_v^r)^*), \tag{21a}$$

$$\text{s.t. } p_v^* > c_v. \tag{21b}$$

Proof: Substituting Eq. (14) and Eq. (17) into Eq. (1), the utility function of the vehicle owner $U_v(p_v, (x_v^r)^*)$ can be transformed into

$$U_v(p_v, (x_v^r)^*) = (p_v - c_v)\left(\frac{\alpha Q(d_v, m_v)}{\sqrt{Q(d_v, m_v)(p_v + M(p_v))}} - 1\right). \tag{22}$$

Taking the second-order derivative of $U_v(p_v, (x_v^r)^*)$ with respect to p_v, we can obtain inequality (23), shown at the bottom of this page. It indicates that $U_v(p_v, (x_v^r)^*$ is a concave function of p_v, satisfying Eq. (6). Therefore, we can derive the optimal strategy p_v^* for the vehicle owner. This completes the proof. □

Hence, the following theorem is derived.

Theorem 1 *The tuple $(p_v^*, (p_v^r)^*, (x_v^r)^*)$ determined by Lemma 1, Lemma 2, and Lemma 3 forms a unique SE in the three-stage Stackelberg game, which represents a strategically stable state in the DVSS framework.*

$$\begin{aligned}\frac{\partial^2 U_v((x_v^r)^*, p_v)}{\partial p_v} &= -\frac{3(a+1)^2\alpha^3 Q(d_v, m_v)^3(c - p_v)}{4(\alpha Q(d_v, m_v)(a x_v^r + b + p_v))^{5/2}} - \frac{(a+1)\alpha^2 Q(d_v, m_v)^2}{(\alpha Q(d_v, m_v)(a p_v + b + p_v))^{3/2}} \\ &= -\frac{(a+1)((a+1)(3c + p_v) + 4b)\sqrt{\alpha Q(d_v, m_v)(a p_v^r + b + p_v)}}{4(a p_v + b + p_v)^3} < 0.\end{aligned} \tag{23}$$

Algorithm 1 Optimal Pricing Mechanism

Input: $p_v(0), p_v^r(0), x_v^r(0), a, b, \alpha, \beta_1, \beta_2, d_v, m_v, k = 0, \delta = 1, \epsilon = 0.0001.$
Output: $p_v^*, (p_v^r)^*, (x_v^r)^*$
1: **while** $\delta > \epsilon$ **do**
2: The renter r adopts strategy to get a higher utility: $x_v^r(k+1) = \frac{\alpha Q(d_v, m_v)}{p_v^r(k)} - 1$,
3: After the renter adopts its strategy, the SP adjusts its pricing strategy as: $p_v^r(k+1) = \sqrt{\alpha Q(d_v, m_v)(p_v(k) + M(p_v(k)))}$,
4: The vehicle owner adjusts its reward strategy as: $p_v(k+1) = \arg\max_{p_v} U_v(p_v(k), x_v^r(k+1))$,
5: $k = k+1$,
6: $\delta = |x_v^r(k+1) - x_v^r(k)|$,
7: **end while**
8: **return** $p_v(k), p_v^r(k), x_v^r(k)$

5.1.2 Algorithm Proposal

Based on the above analysis, we propose Algorithm 1 to obtain the optimal strategies in the three-stage Stackelberg game.

The proposed algorithm iteratively refines the pricing mechanism to achieve optimal strategies for the DVSS. It begins with an initialization phase, setting the initial values for pricing parameters, constants, and iteration variables. The iterative process involves strategic adjustments by the renter, SP, and vehicle owner, aiming to maximize their respective utilities and achieve equilibrium. The renter's strategy is updated based on Eq. (14), capturing the impact of vehicle usage time and mileage. Subsequently, the SP adjusts its pricing strategy using Eq. (17), reflecting considerations of operating costs. The vehicle owner then optimizes its reward strategy by maximizing the utility function with Eq. (21) to align with the overall system dynamics.

The algorithm continues these iterations until the change in the renter's strategy falls below a predefined threshold. The output comprises the final optimal pricing strategies (i.e., p_v^*, $(p_v^r)^*$), and the renter's optimal request strategy (i.e., $(x_v^r)^*$). This iterative approach ensures that the three parties reach an SE, where no party has an incentive to unilaterally deviate from its optimal strategy.

5.2 Three-Sided Stable Matching Game

5.2.1 Definition of TMSC

Three-sided relationships, seen in scenarios like supplier-firm-buyer dynamics and kidney exchange problems, can be effectively modeled in a three-dimensional framework called the Three-Dimensional Stable Marriage problem [26], [27]. This is an extension of the Stable Marriage (SM) model [31], where the three types of matching agents are metaphorically represented as men, women, and dogs, as illustrated in Fig. 2.

The Three-Dimensional Stable Marriage problem encompasses two models based on the nature of agents' preference lists. In the first model, each agent ranks pairs of other agents they are willing to form triples with, shaping their preferences. The second model features preference lists containing only one type of agent. For example, men rank women in the order of preference, women's lists contain only dogs, and dogs rank only men. This specific case is also known as the Three-sided Matching game with Size and Cyclic preference list (TMSC) problem [27] [41].

In the context of the matching game $\mathcal{G}$, we define the set $\mathcal{D} = \mathcal{M} \times \mathcal{V} \times \mathcal{R}$ as the collection of all possible triples. Thus, any matching $\mathcal{D} \subseteq \mathcal{G}$ represents a set of triples selected from $\mathcal{G}$. To identify stability in TMSC, we first need to introduce the concept of a blocking triple.

Definition 2 ***Blocking Triple in TMSC****. A triple $(m, v, r) \notin \mathcal{D}$, but $(m, v, r) \in \mathcal{G}$ is a blocking triple if there exists such a set:*

$$\begin{aligned} &\{\mathcal{D}(m) = \emptyset \vee v \succ_m \mathcal{D}(m)\} \wedge \\ &\{\mathcal{D}(v) = \emptyset \vee r \succ_v \mathcal{D}(v)\} \wedge \{N(m, v) \leqslant N_m\}, \end{aligned} \tag{24}$$

in which $v \succ_m \mathcal{D}(m)$ indicates that SP m prefers vehicle v to its current matched vehicle $\mathcal{D}(m)$. Similarly, $r \succ_v \mathcal{D}(v)$ represents that vehicle v prefers renter r to its current matched $\mathcal{D}(v)$. $N(m, v) \leqslant N_m$ denotes that the total matching amount of vehicle v should not exceed the service capacity of SP m.

In our scenarios, a blocking triple in TMSC consists of an SP, a vehicle, and a renter, where each of them has a preference to be matched with each other as a triple, rather than staying in their current matched partners according to the matching $\mathcal{G}$. A matching $\mathcal{G}$ is deemed stable when no blocking triple exists for $\mathcal{G}$.

5.2.2 Model Formulation

Given the NP-hard nature of solving the optimization problem presented in Eq. (13), we reformulate it as a TMSC model. In the context of our DVSS scenarios, we introduce the following assumptions: SPs rank vehicle owners only, vehicle owners rank renters only, and renters rank SPs only in their respective orders of preferences. Each agent is allowed to be matched with a limited number of agents of the other types, based on their preference rankings. In light of [27], the three-sided matching problem of vehicle sharing is to find a matching $\mathcal{G} = (m, v, r)$ with the maximum cardinality:

$$\max \quad |\mathcal{G}| \tag{25a}$$
$$\text{s.t.} \quad N(\mathcal{G}, m) \leqslant N_m, \forall m \in \mathcal{M}, \forall v \in \mathcal{V}, \forall r \in \mathcal{R}, \tag{25b}$$
$$N(\mathcal{G}, v) \leqslant 1, \forall m \in \mathcal{M}, \forall v \in \mathcal{V}, \forall r \in \mathcal{R}, \tag{25c}$$
$$N(\mathcal{G}, r) \leqslant 1, \forall m \in \mathcal{M}, \forall v \in \mathcal{V}, \forall r \in \mathcal{R}, \tag{25d}$$

where $N(\mathcal{G}, m)$, $N(\mathcal{G}, v)$, and $N(\mathcal{G}, r)$ represent the number of partners that SP m, vehicle owner v, and renter r have in the matching $\mathcal{G}$, respectively. Expression (25a) defines the cardinality of the matching $\mathcal{G}$ (i.e., the number of (m, v, r) triples in the matching). Constraint (25b) imposes a constraint based on the maximum service capacities N_m of SP. Constraints (25c) and (25d) specify that each vehicle and each renter can only be matched with one partner.

According to [8], determining the existence of a stable matching in a TMSC is NP-complete. To address this challenge, we introduce two restrictions into the preference lists of TMSC agents, transforming the problem into the Restricted Three-sided Matching with Size and Cyclic Preference (R-TMSC) model.

- **R-1**: SPs derive their preference lists for vehicles from a master list, which includes all vehicles in strict order. SP preference lists are then derived from this master list, potentially including all or only a subset of the vehicles.
- **R-2**: Renters are indifferent towards SPs, meaning that, for each renter, the SPs in its preference list are considered equally preferable, forming ties.

The TMSC model satisfying the above two restrictions is referred to as the R-TMSC model. However, it is important to note that even in the R-TMSC model, identifying the maximum cardinality matching remains an NP-hard problem, as demonstrated in [8].

5.2.3 Algorithm Proposal

We proceed to devise a three-sided matching algorithm for addressing the R-TMSC problem within the DVSS. Specifically, we establish preference lists for each SP, vehicle (regardless of owner), and renter. In tackling the cyclic preference issue, the preference lists for each agent exclusively encompass counterparts of a different type. Therefore, SPs' preference lists comprise solely of vehicles, the preference lists of vehicles contain only renters, and the preference lists of renters are exclusively composed of SPs. These preference lists are meticulously ordered according to the agents' preferences.

Based on **R-1**, we formulate the preference lists of SPs from a master list that ranks vehicles based on their utility to the SPs, taking into account Equation (11). Higher utility corresponds to a higher preference, as it results in increased royalty income. In our scenario, all SPs share identical preference lists, indicating that all vehicles are equally acceptable to them. Therefore, the preference lists of SPs can be represented as follows:

$$PL_m(v,r) = U_m(v,r), \text{ for } \forall v \in \mathcal{V} \text{ and } \forall r \in \mathcal{R}. \tag{26}$$

The preference lists for vehicles are determined by ranking the acceptable renters based on the vehicle revenue from the renter, as indicated by Equation (12). We denote the preference lists for vehicles as follows:

$$PL_v(r,m) = U_v(r,m), \text{ for } \forall r \in \mathcal{R} \text{ and } \forall m \in \mathcal{M}. \tag{27}$$

Based on **R-2**, the renters are indifferent towards the SPs, reflecting a preference list where all SPs are ranked equally. This can be represented as follows:

$$PL_r(m,v) = 1, \text{ for } \forall m \in \mathcal{M} \text{ and } \forall v \in \mathcal{V}. \tag{28}$$

After generating preference lists for all agents, we define the following sets before introducing our matching algorithm.

$$A^{+1}(\mathcal{G},m) = \{v | v \succ_m \mathcal{G}(m), v \in PL_m(v,r)\}, \tag{29}$$

represents the set of all vehicles preferred by SP m over its current matching $\mathcal{G}(m)$.

$$A^{+1}(\mathcal{G},v) = \{r | r \succ_v \mathcal{G}(v), r \in PL_v(r,m)\}, \tag{30}$$

represents the set of all renters that vehicle v prefers over its current matching $\mathcal{G}(v)$.

$$A^{-1}(\mathcal{G},m) = \{r | r \in \mathcal{R}, m \in PL_r(m,v), N(\mathcal{G},r) \leqslant 1)\}, \tag{31}$$

represents the set of all renters that can still be served by SP m.

$$A^{-2}(\mathcal{G},m) = \{v | v \in \mathcal{V}, A^{+1}(\mathcal{G},v) \cap A^{-1}(\mathcal{G},m) \neq \emptyset\}, \tag{32}$$

indicates that for vehicle v, there is a renter r that vehicle v prefers over its current partner $\mathcal{G}(v)$, and renter r can still be served by SP m.

The proposed three-sided matching algorithm for DVSS is outlined in Algorithm 2. To initiate the matching process, each SP, vehicle, and renter generates its preference lists based on the utility function. In general, an SP m is randomly selected to start, and it chooses the first vehicle v in its preference list PL_m. Subsequently, the selected vehicle v picks the first renter in its preference list PL_v. The indicator $flag$ is used to control the execution of the algorithm, while $\mathcal{G}$ is initially set to empty. The algorithm then aims to search for the best triple and adds this triple to the matching $\mathcal{G}$ at each iteration.

Algorithm 2 Three-Sided Matching Algorithm for DVSS

Input: $\mathcal{M}, \mathcal{V}, \mathcal{R}$
Output: $\mathcal{G}$

1: Construct the preference list PL_m, PL_v, PL_r;
2: Set $\mathcal{G} = \emptyset$, $flag = 1$;
3: **while** $flag == 1$ **do**
4:   $flag = 0$;
5:   **for** each $m \in \mathcal{M}$ **do**
6:     $\mathcal{V}' = A^{+1}(\mathcal{G},m) \cap A^{-2}(\mathcal{G},m)$;
7:     **if** $\mathcal{V}' \neq \emptyset$ **then**
8:       $v = Head(\mathcal{V}', m)$;
9:       $\mathcal{R}' = A^{+1}(\mathcal{G},v) \cap A^{-1}(\mathcal{G},m)$;
10:       $r = Head(\mathcal{R}', v)$;
11:       **if** $N(\mathcal{G},m) == N_m$ **then**
12:         Choose the worst matching for m in set $\{(m, \mathcal{G}(m), \mathcal{G}(\mathcal{G}(m)))\}$;
13:         $\mathcal{G} = \mathcal{G} \backslash \text{worst } \{(m, \mathcal{G}(m), \mathcal{G}(\mathcal{G}(m)))\}$;
14:         Set $flag = 1$;
15:       **end if**
16:       **if** $N(\mathcal{G},v) == 1$ **then**
17:         $\mathcal{G} = \mathcal{G} \backslash \{(*, v, \mathcal{G}(v)\}$;
18:         Set $flag = 1$;
19:       **end if**
20:       $\mathcal{G} = \mathcal{G} \cup (m, v, r)$;
21:     **end if**
22:   **end for**
23:   **if** the unmatched vehicles $\neq \emptyset$ **then**
24:     **if** $N(\mathcal{G},m) < N_m$ & $N(\mathcal{G},r) \leqslant 1$ **then**
25:       Set $flag = 1$;
26:     **end if**
27:   **end if**
28: **end while**
29: **return** $\mathcal{G}$.

In Algorithm 2, the procedure outlined in steps 6-8 is focused on identifying a superior vehicle v for SP m. Here, $Head(X, z)$ represents the element in X with the highest priority in the preference list of z. If a better vehicle v is discovered, the subsequent steps 9-10 are dedicated to the search for an improved renter r for v. The algorithm then proceeds to examine the service capacities of SP m in steps 11-15. Upon reaching the maximum capacities (i.e., $N(\mathcal{G},m)$ equals N_m), the worst matching triple $(m, \mathcal{G}(m), \mathcal{G}(\mathcal{G}(m)))$ within $\mathcal{G}$ is targeted for removal. Similarly, steps 16-19 address the evaluation and processing of the size of vehicle v. Subsequently, the algorithm augments $\mathcal{G}$ with the newly formed triple (m, v, r). Steps 23-27 are designed to ascertain if there are still unmatched vehicles when both SPs and renters possess the capability to match additional vehicles. This iterative process continues until the maximum cardinality $|\mathcal{G}|$ is attained.

5.2.4 Proof of Stability

Algorithm 2 aims to achieve a stable matching among SPs, vehicles, and renters in the context of DVSS. It needs to ensure the convergence to a stable matching within a finite number of iterations. We provide detailed proof of this claim in Theorem 2.

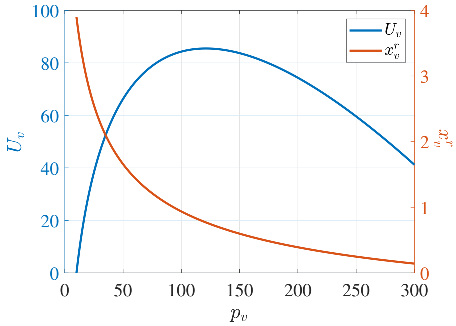


Fig. 3. The values of x_v^r and U_v under different p_v.

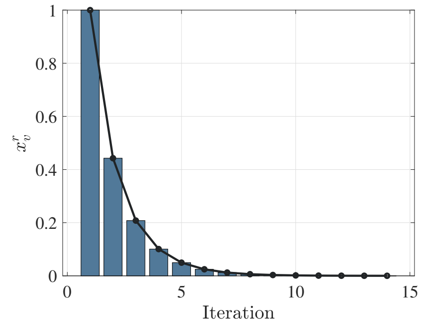


Fig. 4. The value of x_v^r under different numbers of iterations k.

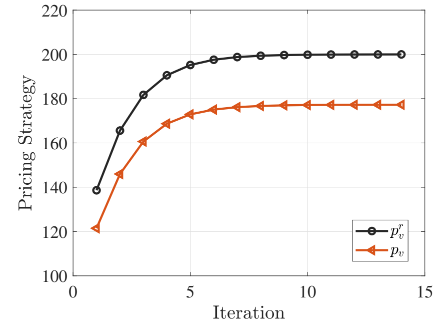


Fig. 5. The pricing strategy under different numbers of iterations k.

Theorem 2 *The three-sided matching algorithm for DVSS is guaranteed to achieve stable matching within a finite number of iterations.*

Proof: In Algorithm 2, the while loop terminates when the flag equals 0. Each iteration in steps 9-20 ensures that a vehicle v is matched to a better renter in its preference list. If a matched vehicle is deleted in step 17, a higher-priority vehicle is matched to a better renter in step 20. Since the number of vehicles and the number of renters in each vehicle's preference list are limited, the iteration will eventually terminate after a finite number of steps.

The stability of the final matching produced by Algorithm 2 is proven through contradiction. Suppose the final matching is not stable, indicating the existence of a blocking triple. This implies that there is an SP m for which a vehicle v in PL_m prefers $\mathcal{G}(m)$ over its current match, and there is a renter r in PL_v preferred by v over its current match, and r can be served by SP m. In other words, $A^{+1}(\mathcal{G}, m) \cap A^{-2}(\mathcal{G}, m) \neq \emptyset$ and $A^{+1}(\mathcal{G}, v) \cap A^{-1}(\mathcal{G}, m) \neq \emptyset$. However, Algorithm 2 will not terminate in such cases. Therefore, the resulting matching from Algorithm 2 achieves stability. □

6 Simulation Results

6.1 Experiment Setting

In this section, we conduct a comprehensive performance evaluation of TRIMP, encompassing an optimal pricing mechanism and a three-sided matching algorithm. First, we validate the effectiveness of our pricing mechanism by examining the optimal strategy in a sequential decision-making scenario involving a vehicle owner, an SP, and a renter. Then, we assess the efficacy of our matching mechanism, which operates seamlessly across a diverse set of SPs, vehicles, and renters.

For the pricing mechanism, we set the vehicle cost c_v to be 10 as the default. For the cost coefficients associated with the SP, we set $a = 0.1$ and $b = 5$. Additionally, we define the utility coefficients for the renter as $\alpha = 10$ and for the vehicle as $d_v = 0.5$ and $m_v = 0.1$. Furthermore, we specify β_1 and β_2 as 5 and 1, respectively. We experimentally validate that our TRIMP archives SE within a finite number of iterations, and then examine its performance under different parameter settings.

For the matching mechanism, we conduct simulations in a scenario involving multiple SPs, vehicles, and renters, specifically with $M = 5$, $V = 150$, and $R = 150$. The utility of SPs from vehicles is defined within the range of $[1, 10]$, and similarly, the utility of renters from vehicles spans the interval $[1, 10]$. Furthermore, we impose a constraint on the maximum number of vehicles that a single SP can service, limiting it to 30. To assess the effectiveness of our TRIMP, we conduct a comparative study against four alternative matching schemes: *Random*, *SP-Optimal*, *Vehicle-Optimal*, and *Normal-TSM*:

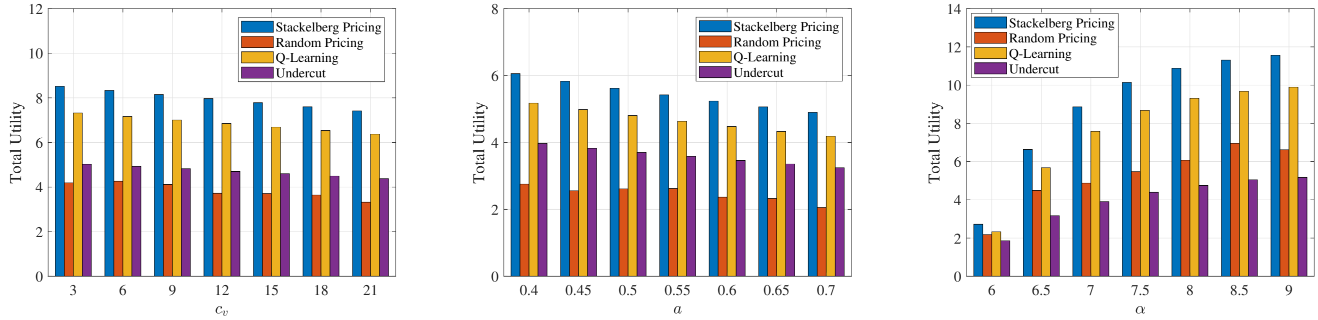
- *Random*: This scheme randomly selects vehicles and renters from the current SP's matching pool, pairing them together.
- *SP-Optimal*: This scheme prioritizes the utility of SPs. It allocates vehicles that maximize the SP's utility to all SPs, with the matched vehicles randomly selected from the pool of renters. This approach aims to enhance the overall satisfaction and utility of SPs within the system.
- *Vehicle-Optimal*: This scheme prioritizes the utility of vehicles. It first selects renters who maximize the utility for all vehicles and then randomly chooses the SP for matching. The goal is to optimize the satisfaction and utility of vehicles within the system.
- *Normal-TSM*: In this scheme, we evaluate Algorithm 2 without the Stackelberg pricing mechanism and refer to it as '*Normal-TSM*'. This approach relies on a preference list generated from the utility values of both the SPs and the vehicles. Subsequently, it utilizes this preference list to facilitate a three-sided matching process.

6.2 Evaluation Results

6.2.1 Convergence Evaluation

Fig. 3 illustrates the relationship between the utility of the vehicle owner v and the renter's request concerning the reward p_v. The concave curve allows us to directly derive the optimal utility for the vehicle owner and the corresponding optimal rental request. Additionally, it is observed that as the reward p_v increases, the renter's request for vehicle v gradually decreases. This sensitivity of the renter's request to the price p_v indicates that higher rewards lead to higher maintenance costs for the SP, resulting in a higher rental price p_v^r and a correspondingly lower customer demand. Furthermore, the utility of the vehicle initially increases and then decreases with the increase of the price p_v. As a result, the vehicle v has an optimal reward strategy p_v^* to maximize its utility. Therefore, the renter decides its optimal request strategy $(x_v^r)^*$ to maximize its utility after the given optimal reward strategy p_v^* and optimal pricing strategy $(p_v^r)^*$.

We investigate the convergence of TRIMP in achieving an SE within a finite number of iterations. The solution for $(x_v^r)^*$ represents the equilibrium point in the Stackelberg game within



(a) (b) (c)

Fig. 6. Total utility comparison under different pricing mechanisms.

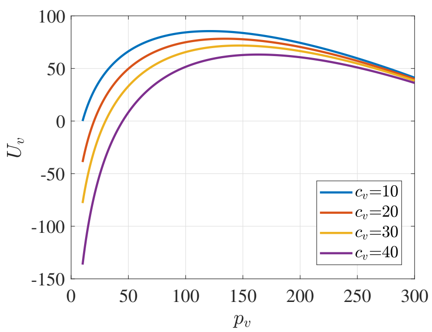


Fig. 7. The value of U_v under different p_v and c_v.

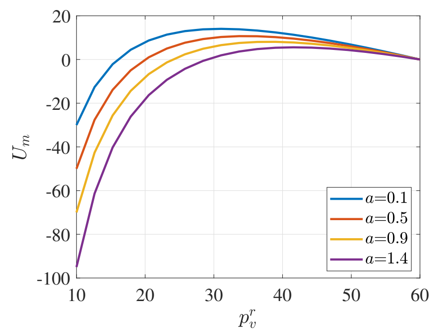


Fig. 8. The value of U_m under different p_v^r and a.

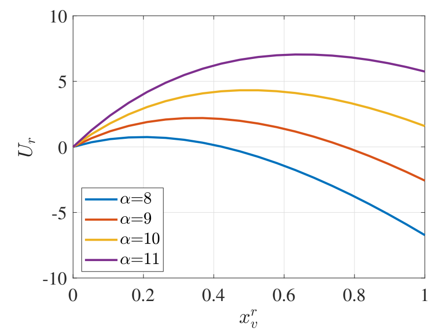


Fig. 9. The value of U_r under different x_v^r and α.

the DVSS. The convergence process of the renter's request is illustrated in Fig. 4. This figure clearly demonstrates a rapid convergence of the renter's request under the initial settings. As the leader adjusts its pricing strategy, the renter exhibits a preference for renting the vehicle to maximize their utility. After approximately 9 iterations, the customer's demand strategy begins to stabilize.

We then analyze the solution $(p_v^*, (p_v^r)^*)$ at the equilibrium point, providing the vehicle owner's optimal reward strategy and the SP's optimal pricing strategy. Fig. 5 illustrates the convergence progress of the vehicle owner and SP. Similar to the convergence of x_v^r, we observe that p_v and p_v^r quickly converge to stable values. The vehicle owner continuously adjusts its optimal reward strategy based on the renter's request to maximize its utility until the strategy no longer changes. Similarly, the SP continuously adapts its pricing strategy in response to the owner's strategy until it stabilizes. This ultimately leads to the formation of a stable equilibrium strategy combination $(p_v^*, (p_v^r)^*, (x_v^r)^*)$ in the three-stage Stackelberg game, as depicted in Fig. 4 and Fig. 5.

6.2.2 Parameter Evaluation

The *Stackelberg* pricing is based on the first decision advantage of the vehicle owner and SP in the DVSS, and the renter's strategy in turn affects the decision-making relationship between the utility of the vehicle owner and SP to construct a sequential decision game to obtain the optimal equilibrium solution of the three parties. In scenarios where the sequential decision-making dynamics are not considered, we adopt other three pricing mechanisms, *Random*, *Q-Learning* [42], and *Undercut* [42] to compare and evaluate with *Stackelberg* pricing. Among them, *Random* pricing refers to the situation where the vehicles, and SPs, without knowing the rental demand, give the self-perceived optimal pricing based on their own utility information and the previous pricing. In *Q-Learning*-based pricing, the vehicle and SP rely on a Q-table to store past experiences and make optimal pricing decisions based on the current action corresponding to the best Q-value in the Q-table. In *Undercut* pricing, the vehicles set low prices to enhance market competitiveness and attract more SPs. In our experiment, we reduce the optimal unit price of the vehicle by 5%-8% to depict the undercut unit price. To intuitively demonstrate the effectiveness of the proposed *Stackelberg* pricing mechanism, we present Fig. 6 to show the total utility under varying parameters c_v, a, and α, compared with the *Random*, *Q-Learning*, and *Undercut* pricing mechanisms. As observed in Fig. 6, the total utility under the proposed *Stackelberg* pricing mechanism is higher than that of the other three pricing mechanisms.

In Fig. 6(a), we illustrate the impact of the vehicle cost c_v on the total utility of the vehicle, SP, and renter in the DVSS, while keeping other parameters constant. It is evident that as the cost of the vehicle rental increases, the total utility under the four pricing methods gradually decreases. This is because as the cost of the vehicle increases, the reward p_v of the vehicle owner also increases in our TRIMP scheme. Consequently, the increase in the rental price p_v^r of the SP leads to a reduction in vehicle demand. Therefore, the total utility of the system decreases marginally with the increase in vehicle cost. The reason why the increase in c_v causes a faster decrease in total utility under *Random* pricing is that there is still a certain difference between the *Random* pricing strategy and the optimal strategy. Consequently, the total utility experiences a notable reduction compared to the *Stackelberg* pricing and is more affected by the increase in the vehicle cost c_v, resulting in a faster decrease in total utility.

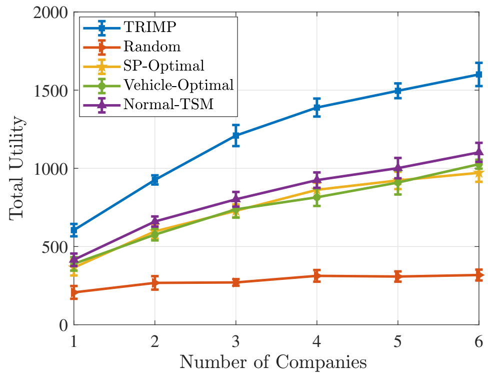


Fig. 10. The total utility under different matching strategies with varying the number of SPs.

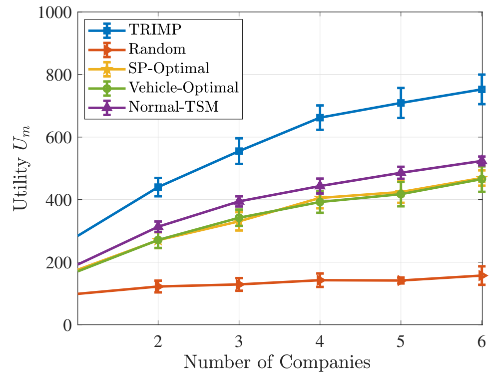


Fig. 11. The utility of SPs under different matching strategies with varying the number of SPs.

Fig. 6(b) shows the impact of the SP service cost coefficient a on the total utility of the vehicle, SP, and renter in the DVSS under different pricing mechanisms. It is evident that the total utility under *Stackelberg* pricing decreases with the increase of a. The observed trend can be primarily attributed to the parameter a serving as the cost coefficient in the utility function of SP. As a increases, the operational cost for SP rises, consequently causing an increase in the rental price p_v^r offered to the renter in our TRIMP scheme. This, in turn, results in a decrease in the renter's demand x_v^r, subsequently leading to a reduction in the overall utility in the DVSS for the vehicle owner, SP, and renter.

Fig. 6(c) is depicted to present the influence of renter satisfaction, represented by the parameter α, on the overall utility of the DVSS. Contrary to the impacts of the vehicle cost and SP cost coefficient, we can observe that as α increases, the utility under *Stackelberg* pricing and *Random* pricing also increases. This is attributed to the fact that, while keeping the other system parameters constant, an escalation in α results in an improvement in customer satisfaction with vehicles of the same price. Therefore, there is a slight increase in the overall system utility. Additionally, among the four pricing methods, *Stackelberg* pricing consistently achieves the highest total utility, followed by *Q-Learning*. However, the utility under *Undercut* pricing is lower than that of *Random* pricing. This is because, as the parameter α increases, the renter's utility also increases. Consequently, pricing below the optimal level will not result in greater utility; instead, it will reduce utility.

We vary c_v from 10 to 40 and present Fig. 7 to illustrate the evolving relationship between the vehicle reward p_v and its utility at varying costs. It is discernible that the utility of the vehicle initially ascends and then descends with the increase of the reward p_v. Each utility curve U_v under different vehicle costs has a maximum point, verifying the existence of its effectiveness extreme point. The utility change trend under different costs is basically the same. However, the greater the cost, the smaller the utility value, which is almost consistent with the impact of vehicle cost on the total utility of the DVSS.

For our analysis, we set p_v to be 10 and then vary the cost coefficient a. Fig. 8 is plotted to illustrate the influence of the cost coefficient a and the SP's pricing p_v^r on the SP's utility. It is notable that the fluctuation pattern in the SP's utility remains essentially consistent across the different cost coefficients a. Additionally, the SP's utility increases first and then decreases with the increase of the SP's rental price. In each utility interval, we observe the presence of maximum points, signifying the existence of an optimal strategy for the SP to maximize its utility. Meanwhile, we find that larger cost coefficients for the SP lead to a diminished utility of SP, which is consistent with the impact of a on the total utility. The optimal pricing strategy under optimal utility progressively increases, aligning with the relationship we established between the cost coefficient and the SP's utility.

We set p_v^r to be 40 and then vary α within the range $[8, 9, 10, 11]$. Fig. 9 is illustrated to analyze the influence of the demand and satisfaction coefficient on the utility function of renters. It is observed that the renter's utility initially rises and then falls with increasing values of x_v^r. In each utility curve, we could identify maximum points, indicating that renters can determine the optimal request strategy to maximize their utility after the SP's pricing strategy is established. Furthermore, a larger value of α results in an increased value in the utility curve and a greater optimal demand strategy. This observation is consistent with the impact of α on the total utility. In other words, a higher value of α corresponds to greater satisfaction and an increased optimal request for the renter.

6.2.3 Comparative Evaluation

To evaluate the final matching performance, we vary the number of SPs and compare our TRIMP with four other schemes. We present the mean and variance results after 10 runs. As demonstrated in Fig. 10, the total utility of the DVSS consistently grows across all five schemes as more SPs join the system. This phenomenon stems from the fact that with fewer SPs providing shared service maintenance in the market, there exists a shortfall in satisfying the shared services for numerous vehicles and users. Therefore, as the number of SPs increases, the capacity to maintain shared vehicles and satisfy user demands expands, consequently resulting in a corresponding rise in the total utility of the DVSS. Upon evaluating our TRIMP mechanism and comparing it with alternative methods, it is evident that our mechanism excels in maximizing the overall system utility. In contrast, the *Normal-TSM* matching algorithm falls due to short of the proposed pricing algorithm. The utility achieved through the *SP-Optimal* and *Vehicle-Optimal* approaches is nearly indistinguishable. Meanwhile, the total utility derived from the *Random* method is the lowest, aligning consistently with our initial algorithm theory.

When the number of SPs is varied from 1 to 6, we illustrate the variation in the utility of SPs in Fig. 11. Among all matching

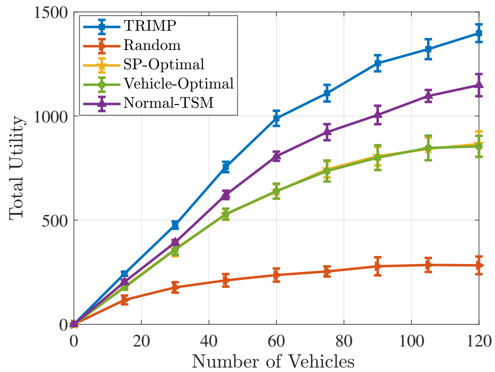


Fig. 12. The total utility under different matching strategies with varying the number of vehicles.

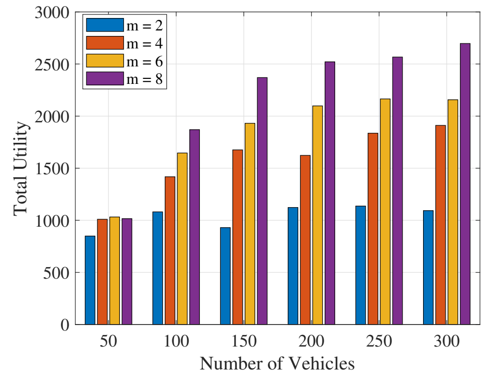


Fig. 14. The total utility under different numbers of SPs with varying the number of vehicles.

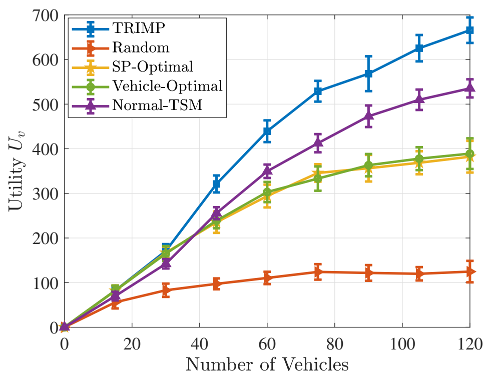


Fig. 13. The balance of SP, vehicle owner, renter, and smart contract.

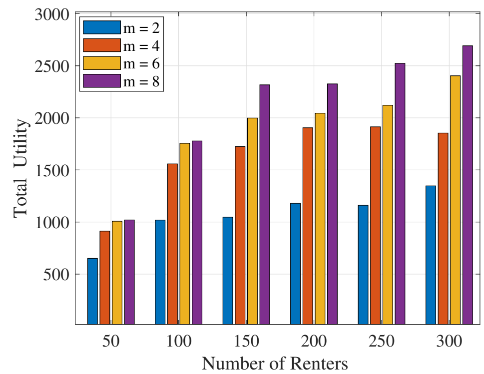


Fig. 15. The total utility under different numbers of SPs with varying the number of renters.

mechanisms, the utility of SPs under our TRIMP significantly outperforms the others, resulting in a notably higher utility. Conversely, the utility growth under *Random* matching is the slowest. Similar to the pattern observed in Fig. 10, it's also evident that the utility of SPs increases with the expansion in the number of SPs across all five matching mechanisms. This is because the increasing number of SPs leads to a larger pool of user services, thereby boosting the overall utility of SPs.

In Fig. 12, we vary the number of vehicles, and observe its impact on the overall system utility for five different matching mechanisms. As the number of vehicles increases to 120, our TRIMP mechanism yielding the highest utility ascends to 1398, while the utility achieved through *Random* matching, which has the lowest utility, only rises to 283. It is apparent that when the SP's service capacity surpasses the number of vehicles, the overall system utility increases positively as the number of vehicles rises. Nevertheless, even with this increase, the TRIMP mechanism is scalable, which continues to exhibit a significant advantage over the total system utility compared to the other four mechanisms.

We illustrate the impact of the number of vehicles on the utility of vehicles in Fig. 13. When the SP's service capacity exceeds the number of vehicles, an increase in the number of vehicles results in more vehicles being matched to both SPs and renters. Consequently, as the number of matched vehicles rises, the utility of vehicles also grows. Simultaneously, it is noteworthy that the relationship between the utility of vehicles and the total system utility remains consistent. This arises from the fact that the relationship between the total utility of vehicles and the proportion of total system utility remains essentially unchanged. Therefore, the total utility of vehicles is directly proportional to the growth of total system utility. Additionally, the utility relationship under the five matching methods is still consistent with the patterns observed in Fig. 10, Fig. 11, and Fig. 12, thus affirming the effectiveness of our TRIMP.

To understand the impacts of varying numbers of vehicles, renters, and SPs on the total utility of the DVSS, we conduct simulations under different parameter settings for vehicle sharing scenarios, and the results are summarized in Fig. 14 and Fig. 15. As shown in both figures, under the operation of the TRIMP mechanism, an increase in the numbers of SPs, vehicles, and renters can result in a gradual increase in the system total utility.

Fig. 14 illustrates that the total utility continues to increase with the rising number of vehicles, regardless of the number of SPs. However, when the number of SPs is 2, the growth trend of total utility slows down with the number of vehicles exceeds 100. This is because, at this point, the number of vehicles surpasses the maximum capacity that 2 SPs can serve, limiting the number of shared vehicles and thus slowing utility growth. When the number of vehicles exceeds 150, even with a fixed number of renters, the availability of more vehicles offers renters more options for matching, leading to a gradual increase in utility.

Similarly, Fig. 15 indicates that as the number of renters increases, the total utility under different numbers of SPs continues to grow. The greater the number of SPs, the higher the total utility. However, when the number of renters is within the range

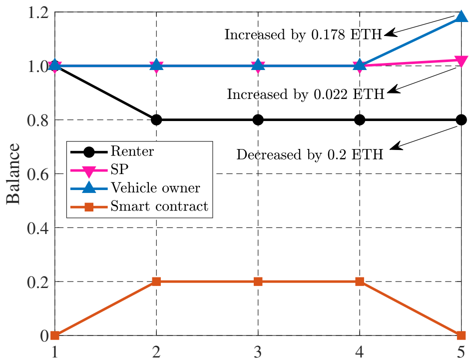


Fig. 16. Balance variation of three parties and smart contract.

of $[50, 100]$, the increase in total utility is less significant if the number of SPs is within the range of $[6, 8]$. This is because a smaller market size means fewer renters can be served, and an increased number of SPs leads to higher operating costs without a proportional increase in utility. When the number of renters is larger, at around 150, the difference in utility becomes more pronounced, demonstrating that a larger market with more customers correlates with a greater total utility.

We implement the DVSS with the Solidity programming language of Ethereum under the compiler version $0.4.26 + commit.4563c3fc$, and the IPFS version is $0.4.14$. Without loss of generality, we only present the interactions of one SP, one renter, and one vehicle in the implementation. Based on the convergence evaluation, the SP can calculate that $p_v^r = 200$ and $p_v = 178$. Due to the preciousness of Ether, we set the unit of the price to be 1 ETH, i.e., 1000 Finney. The implementation steps are as follows.

1) *System Initialization*: The initial account balance for the SP, vehicle, and renter systems are all set as 1 Ether.

2) *Smart Contract Creation*: The renter sets parameters, including p_v^r, rental time, SP address, and agreement, and sends the transaction with p_v^r to the smart contract. At this time, the renter can also call the $abort$ function, which enables canceling the transaction and retrieving the money.

3) *SP passes parameters*: SP calls the $Add\_Vehicle{-}Owner$ function, which can pass the owner's address and p_v. At this point, the renter can still cancel the transaction.

4) *Contract Execution*: SP calls the $Confirmation$ function, which enables the start of the vehicle rental time. At this time, the renter cannot cancel the transaction.

5) *Confirmation*: SP calls the $Check\_if\_end$ function, and if the rental period has ended, the smart contract automatically transfers p_v to the vehicle owner and the remaining funds to the SP. If the rental period has not ended, no action is taken.

We plot Fig. 16 to show the balance variation of the SP, vehicle owner, renter, and smart contract on each implementation step. We can see that through the vehicle sharing transaction, the balance of the renter is decreased by 0.2 Ethers, and the balance of the vehicle owner and SP is increased by 0.178 Ethers and 0.022 Ethers, respectively. The effect of the smart contract is to manage assets temporarily.

7 Conclusion

In this study, we proposed TRIMP, an efficient and stable matching scheme designed for distributed vehicle services. Unlike previous approaches that focused on two-sided matching and pricing, our work addresses the intricacies of a distributed three-sided market involving vehicle owners, users, and SPs. First, to maximize utilities for all three parties involved, we put forward an optimal pricing algorithm grounded in the Stackelberg game framework. This algorithm establishes its equilibrium points through rigorous theoretical proofs and comprehensive experimental validations. Second, leveraging the optimal pricing algorithm, we intricately formulated the matching problem as a three-sided cyclic matching game. We then developed a matching scheme for the dynamic and distributed DVSS environment, and its stability has been rigorously established. Finally, extensive experiments validated the efficacy of TRIMP, demonstrating its capability to achieve higher overall utility in the DVSS compared to four benchmark matching schemes: *Random*, *SP-Optimal*, *Vehicle-Optimal*, and *Normal-TSM*. This showcases its potential to optimize pricing strategies and ensure robust and reliable matching outcomes for distributed vehicle sharing services. In future work, we plan to develop new strategies to enhance the TRIMP framework by addressing fairness concerns and ensuring that the system operates in an equitable and balanced manner for all participants.

References

[1] Z. Ning, S. Sun, X. Wang, L. Guo, S. Guo, X. Hu, B. Hu, and R. Y. K. Kwok, "Blockchain-enabled intelligent transportation systems: A distributed crowdsensing framework," *IEEE Trans. Mob. Comput.*, vol. 21, no. 12, pp. 4201–4217, 2022.

[2] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain security: A survey of techniques and research directions," *IEEE Trans. Serv. Comput.*, vol. 15, no. 4, pp. 2490–2510, 2022.

[3] M. Rodler, W. Li, G. O. Karame, and L. Davi, "Evmpatch: Timely and automated patching of ethereum smart contracts," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 1289–1306.

[4] Y. Duan, X. Zhao, Y. Pan, S. Li, M. Li, F. Xu, and M. Zhang, "Towards automated safety vetting of smart contracts in decentralized applications," in *Pro. ACM SIGSAC Conf. Comput. Commun. Sec.*, 2022, pp. 921–935.

[5] T. Tian, Y. Liu, B. Yan, and J. Li, "Vehicle relocation and staff rebalance problem in one-way free-floating car-sharing system: A novel multi-iteration probabilistic greedy algorithm," *Expert Syst. Appl.*, vol. 238, no. Part A, p. 121528, 2024.

[6] M. Shurrab, S. Singh, H. Otrok, R. Mizouni, V. Khadkikar, and H. H. Zeineldin, "An efficient vehicle-to-vehicle (V2V) energy sharing framework," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5315–5328, 2022.

[7] P. Zhou, C. Wang, and Y. Yang, "Design and optimization of solar-powered shared electric autonomous vehicle system for smart cities," *IEEE Trans. Mob. Comput.*, vol. 22, no. 4, pp. 2053–2068, 2023.

[8] Z. Jia, M. Sheng, J. Li, D. Zhou, and Z. Han, "Joint HAP access and LEO satellite backhaul in 6g: Matching game-based approaches," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1147–1159, 2021.

[9] R. Xie, W. Wei, Q. Wu, T. Ding, and S. Mei, "Optimal service pricing and charging scheduling of an electric vehicle sharing system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 78–89, 2020.

[10] S. Ren, F. Luo, L. Lin, S.-C. Hsu, and X. I. Li, "A novel dynamic pricing scheme for a large-scale electric vehicle sharing network considering vehicle relocation and vehicle-grid-integration," *Int. J. Prod. Econ.*, vol. 218, pp. 339–351, 2019.

[11] T. Kamatani, Y. Nakata, and S. Arai, "Dynamic pricing method to maximize utilization of one-way car sharing service," in *IEEE Int. Conf. Ags., ICA*, 2019, pp. 65–68.

[12] S. Banerjee, D. Freund, and T. Lykouris, "Pricing and optimization in shared vehicle systems: An approximation framework," *Oper. Res.*, vol. 70, no. 3, pp. 1783–1805, 2022.

[13] S. Wang, Y. Yang, Y. Chen, and X. Zhao, "Trip pricing scheme for electric vehicle sharing network with demand prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20 243–20 254, 2022.

[14] J. Pfrommer, J. Warrington, G. Schildbach, and M. Morari, "Dynamic vehicle redistribution and online price incentives in shared mobility systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1567–1578, 2014.

[15] A. Waserhole, V. Jost, and N. Brauner, "Pricing techniques for self regulation in vehicle sharing systems," *Electron. Notes Discret. Math.*, vol. 41, pp. 149–156, 2013.

[16] J. Yang, Y. Lin, F. Wu, and L. Chen, "Subsidy and pricing model of electric vehicle sharing based on two-stage stackelberg game–a case study in china," *Appl. Sci.*, vol. 9, no. 8, p. 1631, 2019.

[17] Z. Peng, W. Shan, P. Jia, B. Yu, Y. Jiang, and B. Yao, "Stable ride-sharing matching for the commuters with payment design," *Transp.*, vol. 47, pp. 1–21, 2020.

[18] S. C.-K. Chau, S. Shen, and Y. Zhou, "Decentralized ride-sharing and vehicle-pooling based on fair cost-sharing mechanisms," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1936–1946, 2022.

[19] H. Zhang and J. Zhao, "Mobility sharing as a preference matching problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2584–2592, 2019.

[20] D. Pelzer, J. Xiao, D. Zehe, M. H. Lees, A. C. Knoll, and H. Aydt, "A partition-based match making algorithm for dynamic ridesharing," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2587–2598, 2015.

[21] G. P. Yatnalka and H. S. Narman, "A matching model for vehicle sharing based on user characteristics and tolerated-time," in *Proc. IEEE 16th Int. Conf. Smart Cities: Improv. Qual. of Life Using ICT (HONET-ICT)*, 2019, pp. 143–147.

[22] X. Wang, N. Agatz, and A. Erera, "Stable matching for dynamic ride-sharing systems," *Transp. Sci.*, vol. 52, no. 4, pp. 850–867, 2017.

[23] S. Rasulkhani and J. Y. Chow, "Route-cost-assignment with joint user and operator behavior as a many-to-one stable matching assignment game," *Transp. Res. Part B: Meth.*, vol. 124, pp. 60–81, 2019.

[24] Y. Xu, M. Xiao, J. Wu, S. Zhang, and G. Gao, "Incentive mechanism for spatial crowdsourcing with unknown social-aware workers: A three-stage stackelberg game approach," *IEEE Trans. Mob. Comput.*, vol. 22, no. 8, pp. 4698–4713, 2023.

[25] B. Qian, H. Zhou, T. Ma, K. Yu, Q. Yuan, and X. Shen, "Multi-operator spectrum sharing for massive iot coexisting in 5g/b5g wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 881–895, 2021.

[26] L. Cui and W. Jia, "Cyclic stable matching for three-sided networking services," *Comp. Netw.*, vol. 57, no. 1, pp. 351–363, 2013.

[27] N. Raveendran, Y. Gu, C. Jiang, N. H. Tran, M. Pan, L. Song, and Z. Han, "Cyclic three-sided matching game inspired wireless network virtualization," *IEEE Trans. Mob. Comput.*, vol. 20, no. 2, pp. 416–428, 2021.

[28] P. Qin, M. Wang, X. Zhao, and S. Geng, "Content service oriented resource allocation for space-air-ground integrated 6g networks: A three-sided cyclic matching approach," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 828–839, 2023.

[29] Z. Miao, Y. Wang, and Z. Han, "A supplier-firm-buyer framework for computation and content resource assignment in wireless virtual networks," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 8, pp. 4116–4128, 2019.

[30] A. Leshem, E. Zehavi, and Y. Yaffe, "Multichannel opportunistic carrier sensing for stable channel access control in cognitive radio systems," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 82–95, 2012.

[31] D. E. Knuth, "Mariages stables et leurs relations avec d'autres problèmes combinatoires: introduction à l'analyse mathématique des algorithmes," *(No Title)*, 1976.

[32] J. Yang, Y. Zhang, Y. Yuan, K. Ma, and J. Li, "Energy trading and power allocation strategies for relay-assisted smart grid communications: A three-stage game approach," *IEEE Trans. Mob. Comput.*, vol. 22, no. 12, pp. 7438–7450, 2023.

[33] M. P. Fanti, A. M. Mangini, G. Pedroncelli, and W. Ukovich, "Fleet sizing for electric car sharing system via closed queueing networks," in *Proc. IEEE Int. Conf. Sys. Mn. Cyber., SMC*. IEEE, 2014, pp. 1324–1329.

[34] M. Nourinejad and M. J. Roorda, "Carsharing operations policies: a comparison between one-way and two-way systems," *Transp.*, vol. 42, no. 3, pp. 497–518, 2015.

[35] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transp. Res. C Emerg. Tech.*, vol. 89, pp. 384–406, 2018.

[36] Z. Wang, B. Gao, and R. Zhang, "Modeling and optimization of the re-balancing problem with re charging requirements in electric vehicle sharing systems," in *Proc. 34th Chin. Ctrl. Decis. Conf. (CCDC)*, 2022, pp. 588–593.

[37] V. Valaštín, K. Košťál, R. Bencel, and I. Kotuliak, "Blockchain based car-sharing platform," in *Proc. Int. Symp. ELMAR*, 2019, pp. 5–8.

[38] M. Kim, J. Lee, K. Park, Y. Park, K. H. Park, and Y. Park, "Design of secure decentralized car-sharing system using blockchain," *IEEE Access*, vol. 9, pp. 54 796–54 810, 2021.

[39] B. Vaidya and H. T. Mouftah, "Security for shared electric and automated mobility services in smart cities," *IEEE Secur. Priv.*, vol. 19, no. 1, pp. 24–33, 2021.

[40] N. Ding, L. Gao, and J. Huang, "Optimal pricing design for coordinated and uncoordinated iot networks," *IEEE Trans. Mob. Comput.*, vol. 22, no. 12, pp. 7121–7137, 2023.

[41] P. Qin, M. Wang, X. Zhao, and S. Geng, "Content service oriented resource allocation for space–air–ground integrated 6g networks: A three-sided cyclic matching approach," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 828–839, 2023.

[42] G. Sun, G. O. Boateng, L. Luo, H. Chen, D. A. Mensah, and G. Liu, "Competitive pricing for resource trading in sliced mobile networks: A multi-agent reinforcement learning approach," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 3830–3845, 2024.

Yang Xu (Senior Member, IEEE) received the B.E. degree from the School of Telecommunications Engineering and the Ph.D. degree from the Department of Communication and Information Systems, Xidian University, Xi'an, China, in 2006 and 2014, respectively, where she is currently an associate professor with the School of Computer Science and Technology. She has published over 50 academic papers at premium international journals and conferences, like the IEEE TRANSACTIONS ON MOBILE COMPUTING (TMC), the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING (TDSC), the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (TWC), etc. Her research interests include wireless communications security, mobile crowd sensing, network economics, block-chain technology, routing protocol design, and so on.

Shanshan Zhang (Graduate Student Member, IEEE) received the M.S. degree from the School of Computer Science and Technology, Xidian University, Xi'an, China, in 2024. Her research interests include intelligence analysis, blockchain, game theory, network economics, and so on.

Chen Lyu (Member, IEEE) received the BS and MS degrees in Telecommunications Engineering from Xidian University of China, Xi'an, China, in 2007 and 2010, and the PhD degree in the Department of Computer Science and Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2016. She is currently an Associate Professor in the Department of Computer Science and Technology at Shanghai University of Finance and Economics, Shanghai, China. Her research interests include wireless security, mobile computing, and security and privacy in online social networks.

Jia Liu (Senior Member, IEEE) received the B.E. degree from the School of Telecommunications Engineering, Xidian University, Xi'an, China, in 2010, and received the Ph.D. degree from the School of Systems Information Science, Future University Hakodate, Japan, in 2016. His research interests include wireless systems security, space-air-ground integrated networks, Internet of Things, 5G, etc. He has published over 80 academic papers at premium international journals and conferences, like the IEEE TRANSACTIONS ON MOBILE COMPUTING (TMC), the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING (TDSC), the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (TWC), the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS), etc. He received the 2016 and 2020 IEEE Sapporo Section Encouragement Award.

Tarik Taleb (Senior Member, IEEE) received the B.E. degree (with distinction) in information engineering and the M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively. He is currently a Full Professor at Ruhr University Bochum, Germany. He was a Professor with the Center of Wireless Communications, University of Oulu, Oulu, Finland. He is the founder of ICTFICIAL Oy, and the founder and the Director of the MOSA!C Lab, Espoo, Finland. From October 2014 to December 2021, he was an Associate Professor with the School of Electrical Engineering, Aalto University, Espoo, Finland. Prior to that, he was working as a Senior Researcher and a 3GPP Standards Expert with NEC Europe Ltd., Heidelberg, Germany. Before joining NEC and till March 2009, he worked as an Assistant Professor with the Graduate School of Information Sciences, Tohoku University, in a lab fully funded by KDDI. From 2005 to 2006, he was a Research Fellow with the Intelligent Cosmos Research Institute, Sendai. Taleb has been directly engaged in the development and standardization of the Evolved Packet System as a member of the 3GPP System Architecture Working Group. His current research interests include AI-based autonomous network management, network softwarization and slicing, mobile cloud networking, architectural enhancements to mobile core networks, network function virtualization, software-defined networking, software-defined security, and mobile multimedia streaming.

Norio Shiratori (Life Fellow, IEEE) received his Ph.D. degree from Tohoku University in 1977. He became an Assistant Professor and Associate Professor at the Research Institute of Electrical Communication, Tohoku University, in 1977 and 1984, respectively. In 1990, he was promoted to Full Professor at the School of Engineering, Tohoku University. In 1993, he commenced his role as Full Professor at the Research Institute of Electrical Communication, Tohoku University. In 1997, Prof. Shiratori became a Visiting Professor at UCLA (University of California, Los Angeles). In 1998, he was elevated to the status of IEEE Fellow. In 2004, Prof. Shiratori assumed the role of Japan representative for the International Federation for Information Processing (IFIP). In 2009, he served as the President of the Information Processing Society of Japan.

In 2010, following his retirement from Tohoku University, Prof. Shiratori took on the positions of Professor Emeritus at Tohoku University and Board Member at Future University Hakodate. In the same year, he became the Chair of the IEEE Sendai Section. In 2012, he was appointed as a Professor at the Global Information and Telecommunication Institute, Waseda University. In 2013, Prof. Shiratori assumed the role of Vice Chair of the IEEE Japan Council. Recognizing his significant contributions, he was honored with the title of IEEE Life Fellow in 2017. Since 2017, he has been serving as a Professor at Research and Development Initiative, Chuo University.

Prof. Shiratori has published over 15 books and over 600 refereed papers in computer science and related fields. He is a fellow of the Japan Foundation of Engineering Societies (JFES), the Information Processing Society of Japan (IPSJ), and the Institute of Electronics, Information and Communication Engineers (IEICE). He was a recipient of the Minister of MEXT Award from the Japanese Government in 2016, the Science and Technology Award from the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) in 2009, the IEICE Achievement Award in 2001, the IEICE Contribution Award in 2011, the IPSJ Contribution Award in 2008, the IEICE Honorary Member in 2012, the IPSJ Honorary Member in 2013, IPSJ Memorial Prize Winning Paper Award in 1985, the IPSJ Best Paper Award in 1997, the IEICE Best Paper Award in 2001, the IEEE 5th SCE01 Best Paper Award in 2001, the IEEE ICPADS 2000 Best Paper Award in 2000, and the IEEE 12th ICOIN Best Paper Award.