Received: 27 April 2021 | Accepted: 15 June 2021 | Published on: 19 August 2021

DOI: 10.1002/net.22074

SPECIAL ISSUE ARTICLE

WILEY

A combinatorial auction-based approach for ridesharing in a student transportation system

Chefi Triki[1,2] | **Maziar Mahdavi Amiri**[3] | **Reza Tavakkoli-Moghaddam**[4] | **Mahdi Mokhtarzadeh**[4] | **Vahidreza Ghezavati**[5]

[1]Division of Engineering Management and Decision Sciences, College of Sciences and Engineering, Hamad Bin Khalifa University, Doha, Qatar
[2]Department of Engineering Innovation, University of Salento, Lecce, Italy
[3]Odette School of Business, University of Windsor, Windsor, Ontario, Canada
[4]School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
[5]School of Industrial Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

Correspondence
Chefi Triki, Department of Engineering Innovation, University of Salento, Lecce, Italy.
Email: chefi.triki@unisalento.it

Funding information
Sultan Qaboos University and Qatar National Library

[Correction added on 16-September-2021, after first online publication: Qatar National Library funding statement has been added.]

Abstract

Here, a mixed-integer linear programming model is developed to represent a transportation system of students traveling from/to a university campus. The concept of ridesharing is used and the mechanism of combinatorial auctions is incorporated within a routing-based model. The mathematical model is based on the vehicle routing problem along with appropriate constraints accommodating features that express the auction clearing phase. A hybrid heuristic-based optimization framework, that takes advantage of meta-heuristic algorithms to improve an initial solution, is also developed to solve large-sized instances of the problem. Three meta-heuristics, namely particle swarm optimization, dragonfly algorithm, and imperialist competitive algorithm, are implemented in the proposed framework, whose performances are assessed and compared. Moreover, two improvement heuristic procedures that attempt to improve the outcomes of the foregoing meta-heuristics are proposed and compared as well.

KEYWORDS

combinatorial auctions, hybrid heuristic algorithms, ridesharing, vehicle routing problem, winner determination problem

1 | INTRODUCTION

Transportation of students is an important daily issue. With the increase in population, the number of vehicles in use has increased dramatically, causing many concerns, such as air pollution, traffic congestion, mounting costs, and reduction of fossil fuel resources. One way to rectify these problems is to expand the public transportation network [36]. Another way for reducing the use of single-passenger cars is ridesharing to accommodate people with similar routes and timetables on the same car. Ridesharing has been seriously considered in recent years, along with public transportation. Furuhata et al. [13] had a comprehensive review of various ridesharing systems and methods and gave a brief description and classification of firms providing this service in the world. Kelley [22], Morency [33], Chan and Shaheen [8], and Furuhata et al. [13] defined ridesharing as a transportation mode, in which people having similar routes and traveling timings may share vehicles and prorate traveling costs (e.g., gasoline, tolls, and parking cost). Ridesharing, in contrast to the public mode of transportation, has more flexibility and is more convenient while having a higher speed. Agatz et al. [1] reviewed optimization models and solution approaches employed in dynamic ridesharing.

Ridesharing turns out to be very useful in university environments. Some campuses, especially those located on the outskirts, suffer from the lack of appropriate and efficient public transportation. Therefore, students without a car may require a long time to commute to the campus. However, in some universities, every day a large number of students commute by single-occupant private cars, causing various traffic problems. This is a typical scenario in wealthy countries where one can see at peak hours-long queues of private cars entering or getting out of the university campus, causing long delays for students and staff. Evidently, ridesharing is a feasible solution to mitigate the effect of such congestion.

Here, a mixed-integer linear vehicle routing problem (VRP)-based model for a ridesharing system is proposed for a university environment. The problem is inspired by a real-life application related to a big university located in the Gulf region and characterized by an inefficient public transportation system and wide utilization of private cars. In this environment, students possessing cars and willing to give rides to other students, on their way to the campus or back, are taken into consideration. The proposed scheme enables students to travel more easily while reducing the problems due to the lack of appropriate public transportation as well as avoiding traffic congestion. Shortage of parking space is also a serious issue on some campuses and ridesharing is a possible way to avoid the costly investment required for expanding the parking infrastructures.

McAfee and McMillan [30] defined the auction as “a market institution with an explicit set of rules determining resource allocation and prices based on bids from the market participants.” In an auction process, the auctioneers are those who want to sell their goods or services (items), and the bidders are those who bid to buy the offered items. The items are allocated to the successful bidders based on the auctioneers’ objective function values (OFVs) [45, 47]. In some cases, an auctioneer is someone who wants to buy goods or receive services, and the bidder is someone who wants to sell their goods or services. According to Triki [42], “Combinatorial auctions are a market exchange mechanism, in which a set of items is put out for bidding for which bidders can submit bids on single or multiple items. The bids can also have a combinational nature, defining thus the so-called bundles of items, allowing the bidders to win the whole bundle or nothing.” By solving the well-known winner determination problem (WDP), the winning bids are identified and the items are assigned to the successful bidders based on maximization/minimization criteria [14, 23].

To the best of our knowledge, there is only one single recent study that deals with the WDP in the context of ridesharing with combinatorial auctions (CAs). Indeed, Karamanis et al. [21] developed a WDP that allows the user to identify the riders–drivers assignment and suggested a robust trip pricing approach. However, this latter work focuses only on the matching-pricing aspects and does not include the routing modeling features to define an integrated framework, as we do in this study.

Here, we make use of the concept of CAs, in which students looking for a ride can bid on being served either in the morning from home to campus (i.e., inbound), or in the evening from campus to home (i.e., outbound). Students can also bid on being served both ways as a bundle (i.e., they should be served either at both rounds or none of them). The auctioneers are students (drivers), who have cars and are willing to participate in the ridesharing system and give rides to other students on their way to the campus or back. The bidders are students (riders) who do not drive a car to the campus and offer a price of being served. The higher the price offered by a rider, the higher his chance of being served by one of the participant drivers.

It is worthy to note that some students may have a car but prefer not to drive to the campus for any reason (e.g., the traffic congestion), and thus they decide to be riders. These riders should, however, guarantee the availability of both inbound and outbound rides to decide not to use their cars. The objective function we elected for our ridesharing system is to maximize the total profit of the system (i.e., the total amount paid by all riders minus the total transportation cost of all the drivers). In this regard, a sensitivity analysis is conducted in Section 5 to analyze the cost imposed by riders not being served. Several not-for-profit objective functions have been considered in the literature and many of them can be adopted here too. These may include, maximizing the social welfare, the number of riders served by the ridesharing system, or the users’ preferences.

Likewise, the objective functions can consist of minimizing the total deviation time for the drivers, the total traveling time for riders or the vehicle miles traveled). However, any objective that does not involve a profit maximization criterion will not be welcomed by the university management that would like to outsource this service and not to add it as a further burden for its duties. Indeed, in the case of our specific context, neither the central administration nor its satellite technical offices accepted to incubate this time-consuming activity. On the contrary, the university prefers to encourage its staff and graduates to launch and manage lucrative activities based on innovative business projects like ridesharing (see Section 6 in [44]). For this purpose, this paper will follow the approach of Hosni et al. [17] in selecting a profit-maximization objective, despite the possible drawbacks it may produce on the system sustainability and fairness among some riders. However, our model can easily accommodate alternative objective functions and pricing techniques that can improve the sustainability, fairness, and service quality aspects at almost no additional cost in terms of model complexity.

Figure 1 represents a simple illustration of the notions being used in our work. The big white circles represent drivers’ locations while the small circles correspond to the riders’ locations. The yellow and blue circles represent, respectively, the riders who only request rides in the morning and the evening, while the red circles represent those who request a ride for both rounds. The green circles correspond to the riders, who request a ride for both rounds as a bundle (they should be served either at both rounds or none of them). For each rider, an arrow is used to show a car serving him/her with an inbound or an outbound transportation solution. Each student needing service may be served either inbound or outbound or both ways. However, some students may be served neither inbound nor outbound because of the scarcity of resources or a low bidding price. In this model,

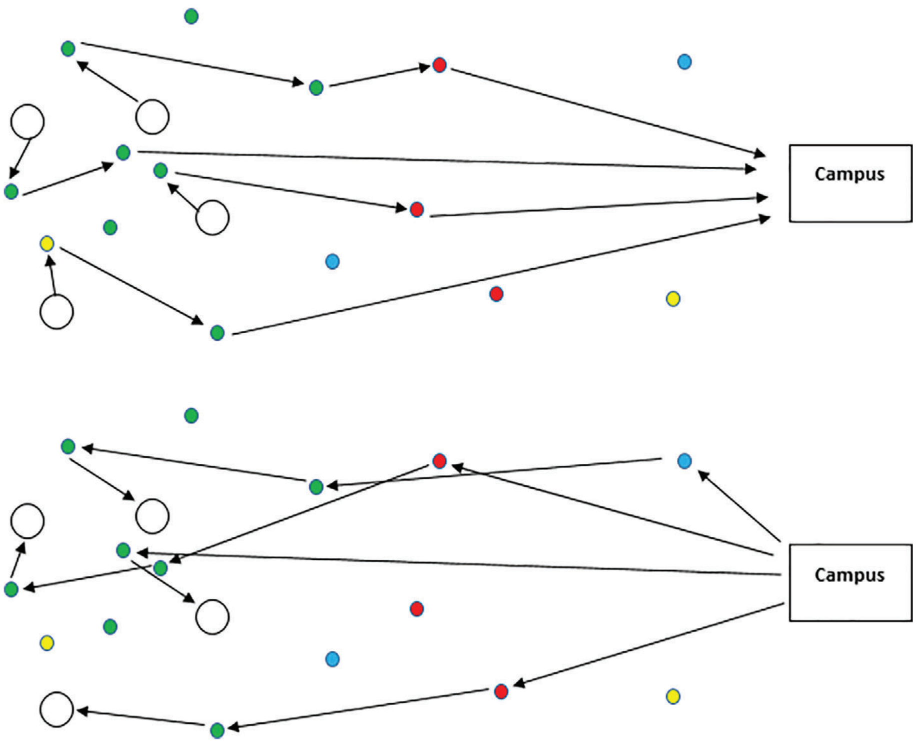


FIGURE 1 Ridesharing network: examples of inbound (up) and outbound (down) trips (uncolored circles are the drivers' locations and colored circles refer to the locations of the different categories of riders: (i) yellow for riders needing rides in the morning, (ii) blue for riders needing rides in the evening, (iii) red for riders requesting rides for both rounds, and (iv) green for riders requesting rides for both rounds as a bundle

riders who are served both in the morning (inbound) and in the evening (outbound) are not necessarily served by the same driver in both rounds.

The contributions of our paper can be summarized as follows:

- We introduce a ridesharing system in a big university campus to reduce traffic congestion and we develop a VRP-based model for its solution;
- We employ for the first time a routing-based CA as a viable mechanism for the staff/students to compete to satisfy their transportation needs;
- We incorporate the WDP within the same integrated optimization model;
- We develop a novel solution framework and suggest an original solution representation characterizing the VRP-WDP model features;
- We propose three meta-heuristic algorithms including particle swarm optimization (PSO), dragonfly algorithm (DA), and imperialist competitive algorithm (ICA) to improve an initial set of solutions generated by the solution framework;
- We propose and develop and two improvement heuristics to improve the quality of the solutions provided by the meta-heuristics;
- We assess the performance of our approaches by using randomly generated test problems that reflect realistic scenarios of the problem under exam.

The remainder of this paper is organized as follows. The literature review is presented in Section 2. In Section 3, the mathematical model is described. Our solution methods are presented and explained in Section 4. Section 5 deals with the discussion and analysis of our computational results. Finally, concluding remarks are in Section 6.

2 | LITERATURE REVIEW

In recent years, considerable and worthy works have been done in the field of ridesharing. Vehicle routing is an important subject in logistics and operations management and applies to a wide range of problems, including ridesharing. The most related application to our context is the dial-a-ride problem (DARP), in which one generally seeks to arrange the routing and scheduling in a way to serve a large number of service requests at the least cost. An excellent review that summarizes the most important DARP models and the related exact and heuristic algorithms can be found in Ho et al. [16]. In this section, we will review the most related works in the area of ridesharing, as well as on those employing the auction mechanism in the context of logistics and transportation.

Amey [3] provided a data-driven method to examine the potential of ridesharing at an organizational level. The method was undertaken on some of the people, who commuted to the Massachusetts Institute of Technology with the provision of a detailed commuting survey. The author compared a potential implementation of ridesharing with what was happening and made inferences about the relative importance of the travel characteristics versus the importance of human attitudes in the coordination of ridesharing. Erdoğan et al. [11] presented the results of a survey using statistical analysis. The authors conducted a questionnaire on the students and staff of the University of Maryland and gathered information about their transportation needs, behaviors, and tendencies. The students' and staff's willingness to participate in ridesharing as drivers or passengers and the factors influencing their willingness were investigated by means of ordered probit models.

Bruglieri et al. [6] proposed a system, named PoliUniPool, to be implemented in two universities in Italy to enable students and staff to jointly use private cars. A heuristic-based on a guided Monte Carlo method was used to solve the carpooling problem. Considering time windows and car capacity as constraints, the considered objective function was a weighted sum of different terms such as maximizing the number of users being served, minimizing the total route length, and maximizing the users' preferences. Al-Riyami et al. [2] presented a ridesharing optimization model based on set partitioning vehicle routing. The objective function was to minimize the total travel time of cars while serving all the requests. The data were about a group of students traveling to a university campus. Triki [42] designed a decision support system based on solving the matching problem enabling the implementation of ridesharing for students and staff of a university. A two-phase optimization approach was proposed to match riders to drivers considering profit maximization. The authors considered in this study only simple requests without any need for implementing the bundling mechanism.

Yan et al. [50] investigated the relationship among the demands for being a passenger, driver, and solo driver (a driver that drives individually without any passenger) in a ridesharing system using a stochastic ridesharing user equilibrium, which consisted of a mode choice model and a route choice model. The authors used the method of successive weighted averages to solve the problem. Hosni et al. [17] formulated and solved the shared-taxi problem in which one looks for several passengers to use a single joint taxi. The authors proposed a mixed-integer model to assign passengers to taxis and to compute the optimal routes for taxis. The model, that maximizes the profit, was solved using a Lagrangian decomposition approach as well as two heuristics. Nourinejad and Roorda [35] presented a centralized mathematical model to express their intended ridesharing system and introduced a decentralized approach as a way to match passengers and drivers. The allocation of passengers to drivers is based on the acceptance or rejection of requests by the passengers and drivers. Santos and Xavier [40] designed a dynamic system for ridesharing and taxi sharing and modeled the system through mathematical programming. In this system, passengers, as well as personal car owners or taxi drivers, specify their conditions and requirements for participation in the scheme, and then both the allocation of passengers and routing are determined. The amount paid by each passenger is calculated according to the maximum payable amount offered by passengers, and also according to the cost of service per kilometer specified by the drivers. The greedy randomized adaptive search procedure was used to solve the problem to maximize the number of accepted requests and minimize the total amount paid by all passengers.

Lee and Savelsbergh [28] presented a single-objective integer program to satisfy a specific fraction of the customers' demands in a ridesharing system. The system makes use of both ordinary drivers and drivers who are additionally assigned to the system. The objective function is to minimize the total service cost of additional drivers. A heuristic method based on a neighborhood search was used to solve large scale instances. Wang et al. [49] presented a binary programming model for a ridesharing problem having the objective of minimizing the total cost, including the cost of the travel time and the toll cost. It is assumed that cars can pass through special lanes when they have a high occupancy rate, and thus decrease their travel times. The cars also pay fewer toll fees. To solve the model, an insertion technique was used to first build an initial solution and then a taboo search algorithm was used to improve the obtained solution. Stiglic et al. [41] proposed a mathematical model in which ridesharing and public transportation were integrated. In this system, those who have a car pick up those who do not have a car and take them to the public transportation stations. So, people not having a car travel part of their routes by ridesharing and the rest by public transportation. Hou et al. [18] investigated a ridesharing problem that matches riders to drivers and conducts vehicle routing. The authors sought to maximize the average loading ratio of the entire system in the objective function and used a large neighborhood search algorithm to solve the problem. Vu [48] used data mining techniques to design a ridesharing framework to provide solutions for some problems and deficiencies of the common ridesharing systems.

On the other hand, the WDP within CAs is needed for identifying the successful bids and proclaiming the winners. In our work, the WDP is used for selecting the riders who will receive service. Different optimization models for the WDP were proposed by Caplice and Sheffi [7] who defined the WDP in truckload transportation auctions. Multi-objective optimization models were presented by Ignatius et al. [19] who considered the cost, market fairness, and market confidence as criteria for choosing the winning bids. Ledyard et al. [27] and Elmaghraby and Keskinocak [10] developed specialized models for real-life applications and Triki et al. [46] embedded a WDP within a production-distribution planning scheme.

In the context of ridesharing and besides the work of Karamanis et al. [21], Lam [26] developed several WDP models to represent three main scenarios related to the kind of service: splittable, non-splittable, and private transportation service. He applied his models to the autonomous vehicle scheduling of a public transportation system and employed the Vickrey–Clarke–Groves pricing scheme to define the charging mechanism. More recently, Yu et al. [52] generalized Lim's results to the case of multiple transportation requests system and developed a WDP that decides the order of handling the requests and their allocation to the autonomous vehicles.

Stochastic WDP models were also investigated in the literature. For instance, a stochastic model was presented by Ma et al. [29] who incorporated the volumes of the transported goods as a random variable. Remli and Rekik [39] and Zhang et al. [56] used a robust paradigm and considered the uncertainty of shipment quantities in the auction process. A constraint generation and a data-driven approach were used to solve the stochastic WDP defined in the previous two works, respectively. Zhang et al. [53] combined sampling-based methods with Monte Carlo simulation for solving the WDP under uncertainty. Finally, Ben Amor et al. [4] and Remli et al. [38] considered the stochasticity of the shipping volumes, the carriers' capacity and the lead-times.

The above analysis of the scientific literature has highlighted the originality of our approach. This study employs for the first time a routing-based CAs mechanism to solve a ridesharing matching problem and develops an integrated VRP-WDP optimization framework and meta-heuristic-based heuristic approaches for its solution.

3 | PROPOSED MATHEMATICAL MODEL

In this section, our WDP ridesharing model is proposed and explained. The model and its assumptions are explained in Section 3.1. In Section 3.2, the mathematical model is presented and finally a penalty cost for non-serving some of the riders is proposed in Section 3.3.

3.1 | Model description

In this model, we have two service rounds: round 1 (inbound in the morning) and round 2 (outbound in the evening). Riders are defined as students who do not commute to the campus by their cars and are willing to be served through the ridesharing system, whereas drivers are students who commute to the campus by their cars and are willing to participate in the ridesharing system to serve the riders. Each rider can request to be served in the morning, in the evening, or both. The riders can also bid for both inbound and outbound rides as a bundle. It is worthy to note that all riders are not necessarily served; that is, after solving the model, some riders may be unserved. However, the cost of unserved riders can be included in the objective function, as investigated in Section 5.

Table 1 presents the feasible combinations of being served for different types of service requests. The number 1 indicates that the rider being served and the number 0 indicates that the service is not received. As can be seen in Table 1, all types of service requests include a column (0, 0) that represents the non-obligation-to-serve of our ridesharing system.

We consider here a network including the campus location (indicated as 0), the drivers' locations, and the riders' locations that are connected by a set of roads. In this model, it is assumed that the time distance between every two nodes at each round is known and that connection between any location and itself is forbidden. Furthermore, the traveling cost per unit of time is known and is the same for both the rounds and for all drivers. In case of availability of more than one driver/rider in the same location, then that location will be replicated as needed and the distance between each node and any of its replication(s) will be set to zero.

Each driver determines the maximum amount of time they are willing to spend at each round and also specifies the maximum number of riders that he/she wishes to serve at each round. On the other hand, riders will not be allowed in this version of the model to impose a maximum traveling time but can accept or reject the suggested traveling plan resulting from the auction clearing process. However, our model can be easily extended to impose a rider-specified trip time by adding few more constraints, but without adding any further insights to the application. It is worth mentioning that the riders who are served at both rounds are not necessarily served by the same driver at both rounds.

Each rider offers a price that he is willing to pay to be served. Clearly, the higher the price offered by a rider, the more likely they will be served. Each rider offers a price for being served at each one of the rounds that they want to be served. If a rider

TABLE 1 Feasible combinations of receiving riding service (the bold font means that the rider is not served)

	Different types of service request									
Round	Only inbound		Only outbound		Both inbound and outbound, not as a bundle				Both inbound and outbound as a bundle	
1 (morning)	**0**	1	**0**	0	**0**	1	0	1	**0**	1
2 (evening)	**0**	0	**0**	1	**0**	0	1	1	**0**	1

TABLE 2 Sets, parameters and variables

Sets	
D	Set of drivers; $D = \{d_1, d_2, \dots, d_k\}$
L	Set of drivers' locations; $L = \{l_1, l_2, \dots, l_k\}$
$R1$	Set of locations of riders requesting only an inbound ride; $R1 = \{r_1, r_2, \dots, r_{n1}\}$
$R2$	Set of locations of riders requesting only an outbound ride; $R2 = \{r_{n1+1}, r_{n1+2}, \dots, r_{n2}\}$
$R3$	Set of locations of riders requesting two-ways rides separately; $R3 = \{r_{n2+1}, r_{n2+2}, \dots, r_{n3}\}$
$R4$	Set of locations of riders requesting bundled two-ways rides; $R4 = \{r_{n3+1}, r_{n3+2}, \dots, r_{n4}\}$
R	$R = R1 \cup R2 \cup R3 \cup R4$
$\{0\}$	Campus location set, indicated as node 0
B	Set of rounds: {1 (inbound for the morning trip), 2 (outbound for the evening trip)}
Parameters	
C_{db}	Capacity of driver d in round b ($d \in D, b \in B$)
T_{db}	Maximum time which driver d is willing to spend in round b ($d \in D$, $b \in B$)
$v1_i$	Price that rider with location i is willing to pay for being served in round 1 ($i \in R1 \cup R3$)
$v2_i$	Price that rider with location i is willing to pay for being served in round 2 ($i \in R2 \cup R3$)
$v3_i$	Price that rider with location i is willing to pay for being served in both rounds ($i \in R4$)
t_{ijb}	Time distance of node i to node j in round b; ($i,j \in \mathrm{L} \cup \mathrm{R} \cup \{0\}$, $b \in B$)
s_{id}	1 if i is the location of driver d; 0, otherwise ($i \in L, d \in D$)
h	cost of traveling per unit of time
K	Total number of drivers participating in the current ridesharing matching
Variables	
X_{ijmb}	Binary decision variable: 1 if driver m travels from node i to node j in round b; 0, otherwise ($i,j \in L \cup R \cup \{0\}$, $m \in D$, $b \in B$)
U_{ib}	Free decision variable needed for sub-tour elimination constraints; ($i \in R$, $b \in B$)
Income	Positive decision variable showing the total income
Cost	Positive decision variable showing the total cost

requests to be served at both rounds, not as a bundle, they should offer two prices, one for each round, but if the rider requests service for both rounds as a bundle, they should offer a single price for being served at both rounds. The objective function is to maximize the total profit of the system which is the total amount of money to be paid by the served riders minus the total cost of transportation. In the sensitivity analysis in Section 5, we also take into account the cost of not serving the riders as a term in the objective function.

After solving the model, the riders who will be served at each round will be determined and the drivers–riders assignment will be known. Also, the order of picking up the riders by each driver in the morning and the order of dropping off the riders by each driver in the evening will be identified. Finally, the total income and cost of the system will be determined. The following formula is proposed as a possible way for profit sharing between the ridesharing system provider and the drivers who participate in at least one of the two ridesharing rounds:

$$\text{Income of each driver } d = \frac{\alpha \times \text{total income of the system} \times \text{total time driven by driver } d}{\text{sum of the total time driven by all drivers}}.$$

However, many alternative ways can be applied, as discussed for example in the review by Guajardo and Rönnqvist [15] and, for the specific case of ridesharing auctions, in Zhang et al. [54, 55], Yu and Lam [51], and Zheng et al. [57]. According to the above formula, α corresponds to the portion of the system's total profit that is distributed among the drivers, while $(1 - \alpha)$ is the portion allocated to the provider of the ridesharing system (with $0 \leq \alpha \leq 1$).

3.2 | Mathematical model

The notations and definitions of sets, parameters, and variables are given in Table 2.

The mathematical model is defined as follows:

$$\text{Max } Z = \mathit{Income} - \mathit{Cost} \tag{1}$$

$$\mathit{Income} = \sum_{i \in R1 \cup R3} \sum_{j \in R1 \cup R3 \cup R4 \cup \{0\}} \sum_{d \in D} X_{ijd1} v1_i + \sum_{i \in R2 \cup R3} \sum_{j \in R2 \cup R3 \cup R4 \cup L} \sum_{d \in D} X_{ijd2} v2_i + \sum_{i \in R4} \sum_{j \in R1 \cup R3 \cup R4 \cup \{0\}} \sum_{d \in D} X_{ijd1} v3_i, \tag{2}$$

$$Cost = \sum_{i \in L \cup R \cup \{0\}} \sum_{j \in L \cup R \cup \{0\}} \sum_{d \in D} \sum_{b \in B} h * X_{ijdb} t_{ijb}, \tag{3}$$

subject to:

$$\sum_{j \in R1 \cup R3 \cup R4 \cup \{0\}} X_{ijd1} = s_{id}, \quad \forall i \in L, \forall d \in D, \tag{4}$$

$$\sum_{j \in R2 \cup R3 \cup R4 \cup L} X_{0jd2} = 1, \quad \forall d \in D, \tag{5}$$

$$\sum_{i \in R1 \cup R3 \cup R4 \cup L} X_{ijd1} - \sum_{s \in R1 \cup R3 \cup R4 \cup \{0\}} X_{jsd1} = 0, \quad \forall j \in R, \ \forall d \in D, \tag{6}$$

$$\sum_{i \in CMP \cup R2 \cup R3 \cup R4} X_{ijd2} - \sum_{s \in R2 \cup R3 \cup R4 \cup L} X_{jsd2} = 0, \quad \forall j \in R, \ \forall d \in D, \tag{7}$$

$$\sum_{i \in R1 \cup R3 \cup R4 \cup L} \sum_{d \in D} X_{i0d1} = K, \tag{8}$$

$$\sum_{i \in R2 \cup R3 \cup R4 \cup \{0\}} X_{ijd2} = s_{jd}, \quad \forall j \in L, \forall d \in D, \tag{9}$$

$$\sum_{i \in R1 \cup R3 \cup R4 \cup L} \sum_{d \in D} X_{ijd1} \leq 1, \quad \forall j \in R1 \cup R3 \cup R4, \tag{10}$$

$$\sum_{i \in R2 \cup R3 \cup R4 \cup \{0\}} \sum_{d \in D} X_{ijd2} \leq 1, \quad \forall j \in R2 \cup R3 \cup R4, \tag{11}$$

$$\sum_{i \in R1 \cup R3 \cup R4 \cup L} \sum_{j \in R1 \cup R3 \cup R4} X_{ijd1} \leq C_{d1}, \quad \forall d \in D, \tag{12}$$

$$\sum_{i \in R2 \cup R3 \cup R4 \cup \{0\}} \sum_{j \in R2 \cup R3 \cup R4} X_{ijd2} \leq C_{d2}, \quad \forall d \in D, \tag{13}$$

$$\sum_{i \in DL \cup R1 \cup R3 \cup R4} \sum_{j \in R1 \cup R3 \cup R4 \cup \{0\}} X_{ijd1} t_{ij1} \leq T_{d1}, \quad \forall d \in D, \tag{14}$$

$$\sum_{i \in R2 \cup R3 \cup R4 \cup \{0\}} \sum_{j \in DL \cup R2 \cup R3 \cup R4} X_{ijd2} t_{ij2} \leq T_{d2}, \quad \forall d \in D, \tag{15}$$

$$\sum_{i \in R1 \cup R3 \cup R4 \cup L} \sum_{d \in D} X_{ijd1} - \sum_{s \in R2 \cup R3 \cup R4 \cup \{0\}} \sum_{d \in D} X_{sjd2} = 0, \quad \forall j \in R4, \tag{16}$$

$$U_{ib} - U_{jb} + C_{db} X_{ijdb} \leq C_{db} - 1, \quad \forall i, j (i \neq j) \in R, \forall b \in B, \forall d \in D, \tag{17}$$

$$X_{ijdb} \in \{0, 1\}, \quad \forall i, j \in L \cup R \cup \{0\}, \forall d \in D, \forall b \in B, \tag{18}$$

$$U_{ib} \text{ free variable}, \quad \forall i \in R, \forall b \in B, \tag{19}$$

$$Income \geq 0, \tag{20}$$

$$Cost \geq 0. \tag{21}$$

The objective function (1) maximizes the total profit of the system. Formula (2) calculates the income of the system which is the total amount of money paid by the served riders. Formula (3) calculates the total cost of transportation. Constraints (4) indicate that in round 1, all drivers start their tours from their locations only once. Constraints (5) impose that all drivers start their routes from the campus in round 2, only once. Constraints (6) and (7) ensure that in both rounds if a driver enters a rider's location, they will exit the location and vice versa. Constraint (8) establishes that all drivers enter the campus in round 1 only once. Constraints (9) ensure that all drivers enter their locations in round 2. Constraints (10) impose that in round 1, each rider can only be served by one driver only once. Constraints (11) impose that in round 2, each rider can only be served by one driver only once. Constraints (12) and (13) impose that the number of riders who are served by each driver in each round should not exceed the limit specified by drivers. Constraints (14) and (15) establish that the time each driver spends in each round must not exceed the specified time limit. Constraints (16) ensure that each rider requesting both inbound and outbound rides as a bundle, must either be served at both rounds or not be served at any of the two rounds. Constraints (16) are the main constraints regarding the combinatorial nature of the auction in the model. This set of constraints is the main link between round 1 and round 2. Constraints (17) are the well-known Miller–Tucker–Zemlin like subtour elimination constraints. In (17), U_{ib} are continuous free variables that are used to indicate the service order of rider i along round b's route and C_{db} is used here to represent the maximal number of riders that driver d can serve in round b (see [37]). Constraints (18)–(21) define the nature of all variables. The cost function (3) ensures that in the optimal solution, none of the drivers enters drivers' locations in round 1, none of them exits campus in round 1, none of them exits drivers' locations in round 2 and none of them enters campus in round 2.

It is worth noting that such a model will cover the most congested arrival departure timeslots of the day. Indeed, a peculiar characteristic of our case-study shows that even though lectures start/end every 2 h, intensive arrivals to the campus (with extremely long queues in front of the controlled gates) happen at the beginning of the day (i.e., 8:00 a.m.) and intensive departures from the campus happen every afternoon of the weekdays (i.e., at 6:00 p.m.). Arrival/departure traffic at the campus's gates during the remaining hours of the day is very limited and is thus considered practically negligible. A natural extension of the above model is to incorporate time windows in the clearing auction process whenever the arrival/departure activities of the students will intensify along the whole academic day.

3.3 | Penalty cost of unserved riders

Here, a cost function is defined which involves, in addition to the total transportation cost, the cost of unserved riders. This function is to be used in the sensitivity analysis in Section 5, where g is a flat-penalty cost of not serving a rider in one round:

$$\begin{aligned} Cost2 = & \sum_{i\in L\cup R\cup\{0\}} \sum_{j\in L\cup R\cup\{0\}} \sum_{d\in D} \sum_{b\in B} h\, X_{ijdb}\, t_{ijb} + \sum_{i\in R1\cup R3\cup R4} \sum_{j\in R1\cup R3\cup R4\cup\{0\}} \sum_{d\in D} g\,(1 - X_{ijd1}) \\ & + \sum_{i\in R2\cup R3\cup R4} \sum_{j\in R2\cup R3\cup R4\cup L} \sum_{d\in D} g\,(1 - X_{ijd2}). \end{aligned} \tag{22}$$

4 | SOLUTION FRAMEWORK

The proposed model is based on the VRP, which is known to be NP-hard. Furthermore, the complexity of the proposed model is exacerbated by having two rounds and by the link between both of them with riders who request bundled rides. Here, we propose a novel hybrid heuristic algorithm which implements meta-heuristic algorithms to solve large-scale instances. We refer to this hybrid heuristic algorithm as *Combinatorial Auction RideSharing Solution Framework* (*CA-RS*) that involves an original solution format to allow embedding both the routing and auction features within the same solution representation, as well as several other innovative components. This framework implements meta-heuristic algorithms to improve a set of initially generated solutions. Then it applies a local search on the solution of a meta-heuristic algorithm to find the best possible solution. Although most meta-heuristic algorithms can be used in this context, PSO, ICA, and DA are chosen based on our experiences and on the good performance, they have shown while solving similar problems (refer to the following related studies: [12, 20, 25, 31, 32, 34]). For the sake of brevity, the proposed meta-heuristic algorithms are described in the Online Appendices.

4.1 | Combinatorial Auction RideSharing Solution Framework

First, *npar* random initial solutions are generated and converted into feasible solutions (Section 4.2). Next, one of the meta-heuristic algorithms is applied to the feasible initial solutions, with *bpso* being the best solution obtained by the meta-heuristic method (Section 4.3). Subsequently, improvement heuristic techniques, which are referred to as heuristic phases (*Hphases*), are implemented separately on *bpso* and on all *npar* feasible solutions which are at hand after the termination of the meta-heuristic algorithm to obtain the best possible solution, *bfinal* (Section 4.4). All procedures are discussed in detail in the following subsections. Figure 2 illustrates the overall *CA-RS* process.

4.2 | Initial solutions, solution formats

In the sequel, riders requesting only an inbound ride are referred to as type 1 riders, riders requesting only an outbound ride are referred to as type 2 riders, and riders requesting both inbound and outbound rides, separately, are referred to as type 3

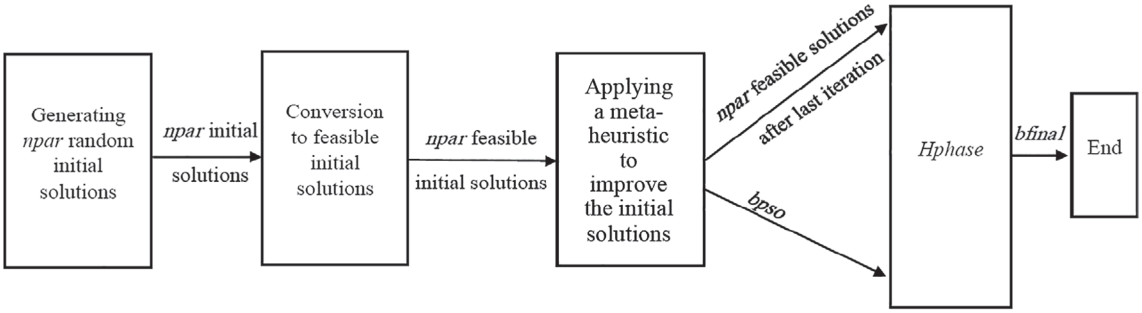
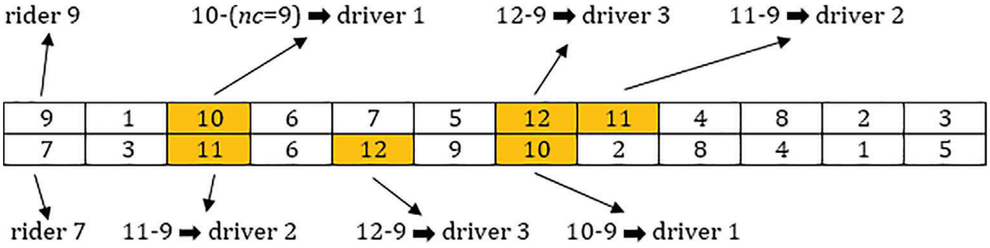

Generating *npar* random initial solutions
npar initial solutions
Conversion to feasible initial solutions
npar feasible initial solutions
Applying a meta-heuristic to improve the initial solutions
npar feasible solutions after last iteration
bpso
Hphase
bfinal
End

FIGURE 2 Overall process of *CA-RS*



9	1	10	6	7	5	12	11	4	8	2	3
7	3	11	6	12	9	10	2	8	4	1	5

FIGURE 3 An example of solution representation

0.14	0.72	0.88	0.57	0.38	0.24	0.31	0.63	0.1	0.2	0.5	0.45
0.89	0.66	0.21	0.77	0.97	0.34	0.15	0.68	0.49	0.55	0.29	0.4

FIGURE 4 Format of a continuous solution

riders. Finally, riders requesting both inbound and outbound rides as a bundle are referred to as type 4 riders. For a problem with nc riders and nv drivers, the solution format is a $2 \times (nc + nv)$ matrix whose first row is for round 1 and its second row is for round 2. The elements of each row are real numbers randomly generated from the interval [0, 1]. The elements of each row are ascendingly sorted to represent a solution. Therefore, each row will be a permutation of 1 to $nc + nv$. An example with 3 drivers ($nv = 3$) and 9 riders ($nc = 9$) is shown in Figure 3 (the solution format of this solution is shown in Figure 4, in which the elements are sorted to obtain the drawn permutation). All numbers greater than nc represent the number of drivers in sequence. The orange elements are drivers, the other elements are riders. Riders 1 and 2 are type 1 riders, riders 3 and 4 are type 2 riders, riders 5 and 6 are type 3 riders, and riders 7–9 are type 4 rides. The elements before each driver element are assumed to be assigned to that driver in the sequel. Note that if two elements are swapped during the process of solution correction, their solution format values are also swapped to reflect the changes in the solution format.

The route that each driver should follow is composed of the riders reported just before the number of that driver in the solution sequence. For example, as shown in Figure 3, the route of driver 1 round 1 includes rider 9, rider 1 and then the campus, whereas in round 2 driver 1's route involves rider 9 and then her/his location. The riders who are at the end of each row of the solution matrix are not served by any drivers. For example, riders 4, 8, 2, 3 are unserved in the morning and riders 2, 8, 4, 1, 5 are unserved in the evening.

npar random initial solutions are generated, and each row is sorted. Then, the $f1$ and $f2$ modifier operators are performed respectively on all the solutions to correct any infeasibility within the solutions, and *npar* feasible solutions are, thus, obtained. Note that the $f2$ operator will be performed on each solution if and only if the solution is infeasible in terms of time limits and/or the limits of the number of riders assigned to each driver.

The $f1$ operator works, as illustrated in Algorithm 1, when performed on a solution to make row 1 of that solution feasible respecting type 2 riders. After applying $f1$ operator, no type 2 rider will be served in the morning. Also, a similar process as described in Algorithm 1 is applied to row 2 of the matrix, and thus no type 1 rider will be served in the evening.

An example will show how $f1$ works. Assume there are 3 drivers ($nv = 3$) and 9 riders ($nc = 9$). Riders 1 and 2 are of type 1, riders 3 and 4 are of type 2, riders 5 and 6 are of type 3, and riders 7–9 are of type 4. If $f1$ is performed on the infeasible solution shown in Figure 5, the green elements are removed and sent to the end of the corresponding rows and the solution shown in Figure 6 is obtained. In Figures 5 and 6, the elements regarding the drivers are shown in orange.

The $f2$ operator works as illustrated in Algorithm 2 when performed on row 1 of a solution to make row 1 of that solution feasible respecting time limits and/or capacity constraints for any driver. Also, Aa similar process described in Algorithm 2 is

1	4	11	7	10	9	2	6	12	8	5	3
11	9	8	3	2	12	4	5	6	10	1	7

FIGURE 5 Infeasible solution in terms of the type of served riders

1	11	10	9	2	6	12	8	5	3	4	7
11	9	3	12	4	5	6	10	1	7	2	8

FIGURE 6 Feasible solution obtained after applying $f1$ on the solution shown in Figure 5

performed on the second row (evening) to make row 2 of that solution feasible. Here, for riders of type 4, the price which the rider proposes for being served at both rounds as a bundle should be considered, but for riders of type 3, the price proposed for being served in each round should be considered independently.

Algorithm 1: Procedure of $f1$ operator

```
1:  $d \leftarrow 0$ \\ counter for the number of drivers that are checked in lines 8–10
2:  for $i = 1 : (nc + nv)$ do
3:    $sr \leftarrow i's$ element of row 1 of the solution matrix
4:    if $sr \le nc$ do \\meaning that $i$ is a rider
5:      if $sr \in R_2$ do
6:        move $sr$ to the end of row 2 in the solution matrix
7:      elseif $sr \in R_4$ do
8:        if $sr$ is served by no driver in row 2 do
9:          move $sr$ to the end of both rows 1 and 2 in the solution matrix
10:       endif
11:     endif
12:   else do
13:     $d \leftarrow d + 1$
14:   endif
15:   if $d = nv$ do
16:     break
17:   endif
18: endfor
```

Algorithm 2: Procedure of $f2$ operator

```
1:  for $d \in D$ do
2:      $t \leftarrow$ total time that driver $d$ spend in round 1
3:      $c \leftarrow$ total number of riders that are assigned to driver $d$ spend in round 1
4:      while $t > T_{d,1} \parallel c > C_{d,1}$ do
5:          $sr \leftarrow$ find a served rider by driver $d$ in rounds 1 with the lowest price
6:          move $sr$ to the end of row 1 in the solution matrix
4:          if $sr \in R_4$ do
5:            move $sr$ to the end of the row 2 in the solution matrix
6:          endif
7:          $t \leftarrow$ total time that driver $d$ spend in round 1
8:          $c \leftarrow$ total number of riders that are assigned to driver $d$ spend in round 1
9:      endwhile
10: endfor
```

4.3 | Meta-heuristic algorithms

After the initial solutions are generated and converted to the initial feasible solutions at the first and second stage of the CA-RS, one meta-heuristic algorithm is applied at the third stage of the *CA-RS* to improve the initial solutions. We propose three meta-heuristics (i.e., PSO, DA, and IC, as described in the online Appendices) to improve the initial solutions. The following steps are applied by any meta-heuristic algorithm to generate new feasible solution at each iteration of that algorithm:

Step 1. Generate a new set of solutions using the operator(s) of the implemented meta-heuristic algorithm. For example, PSO simulates the behavior of birds while they are looking for food to generate new solutions.
Step 2. Apply $f1$ and $f2$ modifier operators on the newly generated solutions to make them feasible.

4.4 | Improvement heuristic techniques (Hphase)

At this stage, $npar + 1$ feasible solutions are at hand, which have been produced in the previous step. A local search that includes six moves, that is, $hr1$, $hr2$, $hr1$, $hr3$, $hr1$, and either $hz1$ or $hz2$ is developed to improve the solutions provided by the previous step of the framework. Steps 1–3 will be consecutively applied to all of these $npar + 1$ solutions. At the end, $npar + 1$ outputs will be at hand and the best of them in terms of the objective function will be selected as the final solution of *CA-RS* (*bfinal*). Steps 1–3 are as follows:

Step 1. Calculate the objective function at the solution and assign its value to the variable called *mo*.
Step 2. Apply the techniques $hr1$, $hr2$, $hr1$, $hr3$, $hr1$, and either $hz1$ or $hz2$ sequentially on the solution.
Step 3. Calculate the objective function of the solution. If the value of the objective function is greater than *mo*, send the solution to step 2 and assign the value of the objective function to *mo*. Otherwise, set the existing solution as the output and terminate the steps.

$hz1$ or $hz2$ are used to improve the objective function. They are removing a number of riders that pay the least prices, and by removing them, there may be the opportunity to repeat step 2 to search for a better solution. If $hz1$ is used in step 2, then we refer to the entire hybrid meta-heuristic-based algorithm as *CA-RS*1, whereas if $hz2$ is applied, then it is referred to as *CA-RS*2. Next, we describe the techniques $hr1$, $hr2$, $hr3$, $hz1$, and $hz2$.

4.4.1 | $hr1$

For each driver at each round, check all different combinations of its riders' getting off or getting on, and select the combination that has the smallest traveled distance. For example, if the driver has four passengers at a round, check $4! = 24$ combinations for that driver at that round.

4.4.2 | $hr2$

Step (1). Consider the solution to which $hr2$ is applied as the current solution. In the current solution, find all unserved riders, at all rounds, and of all types, and sort them based on their offered bids from highest to lowest, and put them in list no. 1. For type 1 and type 2 riders, the price is what they offer for being served at round 1 and round 2, respectively. For type 3 riders, there are two prices considered, one is what they offer for being served in the morning, and one is what they offer for being served in the evening. So, a rider of type 3 may be put in the list no. 1 at most twice, once for not being served in the morning and once again for not being served in the evening. For type 4 riders, the price is what they offer for being served at both rounds. Therefore, each of the unserved riders of type 4 will be put once in list no. 1.

Step (2). Select the first rider of list no. 1 and remove the rider from the list:

- If the rider is of type 1, or is of type 3 and is selected because of not being served in the morning, the element corresponding to the rider will be removed from row 1 of the solution matrix and will be placed on the left end of row 1, and will be exchanged with its adjacent element one by one. At every level, the objective function is calculated. Whenever the first positive change in the objective function is seen, if the solution is feasible, we stop and the solution will be set as the current solution. If the element of the rider becomes adjacent to the last element of a driver in row 1 and still no feasible solution with an improvement in the objective function is found, then we stop and the existing current solution will remain unaltered.
- If the rider is of type 2 or 3 and is selected because of not being served in the evening, the element corresponding to the rider will be removed from row 2 of the solution matrix, placed on the left end of row 2,

and exchanged with its adjacent element one by one. At every level, the objective function is calculated. Whenever the first positive change in the objective function is seen, if the solution is feasible, we stop and the solution will be set as the current solution. If the element of the rider becomes adjacent to the last element of a driver in row 2 and still no feasible solution with an improvement in the objective function is found, then we stop and the existing current solution will remain unaltered.

- If the rider is of type 4, the element corresponding to the rider will be removed from rows 1 and 2 of the solution matrix and will be placed on the left end of rows 1 and 2, and in row 2 the element of the rider will be exchanged with its adjacent element one by one until it becomes adjacent to the last element of a driver in row 2. Then, the element of the rider will be again placed on the left end of row 2 and the element of the rider will be exchanged with its adjacent element in row 1, and then in row 2 the element of the rider will be exchanged with its adjacent element one by one until it becomes adjacent to the last element of a driver in row 2. This process will repeatedly be continued. At every level, the objective function is calculated. Whenever the first positive change in the objective function is seen, if the solution is feasible, we stop and the solution will be set as the current solution. If the element of the rider becomes adjacent to the last element of a driver in both rows 1 and 2 and still no feasible solution with an improvement in the objective function is found, then we stop and the existing current solution will remain unaltered.

Step (3). Repeat step 2 until list no. 1 is empty. The last current solution will be the output.

4.4.3 | *hr*3

Step (1). Consider the solution to which *hr*3 is applied as the current solution.

Step (2). Calculate 20% of the total number of riders (*nc*) and assign its integer part to the *ch* variable.

Step (3). In row 1 of the solution matrix, among the riders who are served in the morning, specify *ch* pairs of the riders having the smallest total time distance in the morning (the total time distance of pair (*a*, *b*) is the time distance from node *a* to node *b* plus the time distance from node *b* to node *a*, where the time distance from node *a* to node *b* means the time that should be spent to drive from node *a* to node *b*). Switch the elements of pair 1 riders in row 1 and then calculate the objective function. If there is an improvement in the objective function and the solution is feasible, set the solution as the current solution; otherwise, keep it unaltered.

Step (4). Perform the process explained in step 3 on the current solution for the rest of the pairs, until reaching the pair *ch*, and update the current solution at every level.

Step (5). In row 2 of the solution matrix, among the riders who are served in the evening, specify the *ch* pairs of the riders having the smallest total time distance in the evening. Switch the elements of pair 1 riders in row 2 and then calculate the objective function. If there is an improvement in the objective function and the solution is feasible, then set the solution as the current solution; otherwise, keep it unaltered.

Step (6). Do the process explained in step 5 on the current solution for the rest of the pairs, until reaching the pair *ch*, and update the current solution at every level. The output will be the last current solution.

4.4.4 | *hz*1

Procedure of *hz*1 is illustrated in Algorithm 3. In this operator, the price for a type 3 rider is what they offer for being served at a round, and the price for a type 4 rider is what they offer for being served at both rounds as a bundle.

Algorithm 3: Procedure of *hz*1 operator

```
$hf \leftarrow 0$
while $hf \leq \lfloor 0.15nc \rfloor$ do
    $sr \leftarrow$ find a served rider in rounds 1 or 2 with the lowest price
    move $sr$ to the end of its corresponding row in the solution matrix
    if $sr \in R_4$ do
        move $sr$ to the end of the other row in the solution matrix
    endif
    $hf \leftarrow hf + 1$
endwhile
```

TABLE 3 Information about examples

Instance	No. of nodes in each round	No. of drivers	No. of riders	No. of type 1 riders	No. of type 2 riders	No. of type 3 riders	No. of type 4 riders
E-1	20	4	15	5	5	0	5
E-2	30	5	24	7	7	0	10
E-3	30	5	24	6	6	6	6
E-4	35	5	29	9	9	0	11
E-5	45	8	36	9	9	9	9
E-6	53	8	44	11	11	11	11
X-1	70	10	59	12	11	14	22
X-2	85	15	69	16	14	18	21
X-3	100	15	84	18	18	18	30
X-4	125	24	100	21	24	25	30
X-5	150	30	119	27	27	27	38
X-6	180	44	134	28	29	32	45
X-7	210	59	150	35	35	40	40
X-8	250	70	179	30	35	45	69
X-9	301	80	200	40	40	60	70
X-10	400	100	299	55	70	80	94

4.4.5 | *hz*2

Procedure of *hz*1 is illustrated in Algorithm 4. Here, the price for a type 3 rider is what they offer for being served at a round, and the price for a type 4 rider is what they offer for being served at two rounds as a bundle.

Algorithm 4: Procedure of *hz*2 operator

```
for $d \in D$ do
    $sr \leftarrow$ find a served rider by driver $d$ in rounds 1 with the lowest price
    move $sr$ to the end of row 1 in the solution matrix
    if $sr \in R_4$ do
        move $sr$ to the end of the row 2 in the solution matrix
    endif
endfor
for $d \in D$ do
    $sr \leftarrow$ find a served rider by driver $d$ in rounds 2 with the lowest price
    move $sr$ to the end of row 2 in the solution matrix
    if $sr \in R_4$ do
        move $sr$ to the end of row 1 in the solution matrix
    endif
endfor
```

In general, using technique *hz*2 is more time consuming as compared to technique *hz*1, because more elements will be deleted using *hz*2 and as a result, more computations are needed during *CA-RS*.

5 | COMPUTATIONAL RESULTS

In this section, our computational results for the heuristic and exact solutions are presented and discussed through some instances. Here, all instances and data are randomly generated in such a way that they reflect realistic scenarios. All prices, income, and cost are based on a monetary unit. The information about all instances is given in Table 3. The distances are based on the Euclidean distance. All computations are performed on a computer with Intel Core i7-3537U, 2.0 GHz CPU, 6 GB RAM and Windows 8 Enterprise. All exactly solved examples are coded in GAMS 24.1.2 software package and solved by the CPLEX solver. The heuristic methods are coded and run on MATLAB R2016b (9.1.0.441655) software environment.

TABLE 4 Prices offered by riders to be served (E-3)

	Rider											
Price	**1*T*1**	**2*T*1**	**3*T*1**	**4*T*1**	**5*T*1**	**6*T*1**	**1*T*2**	**2*T*2**	**3*T*2**	**4*T*2**	**5*T*2**	**6*T*2**
*v*1	171	190	190	209	228	228	—	–	—	—	—	—
*v*2	—	—	—	—	—	—	95	133	171	95	114	190
*v*3	—	—	—	—	—	—	—	—	—	—	—	—
	Rider											
Price	**1*T*3**	**2*T*3**	**3*T*3**	**4*T*3**	**5*T*3**	**6*T*3**	**1*T*4**	**2*T*4**	**3*T*4**	**4*T*4**	**5*T*4**	**6*T*4**
v1	133	152	114	95	171	76	—	—	—	—	—	—
v2	114	152	133	114	171	114	—	—	—	—	—	—
v3	—	—	—	—	—	—	380	418	285	760	760	380

TABLE 5 Capacity of drivers, number of riders served by each driver, time limit, and spent time of each driver (E-3)

	C_1	S_1	C_2	S_2	L_1	T_1	L_2	T_2
*d*1	4	4	4	4	90	83.5	120	95.9
*d*2	2	2	2	2	85	62	130	85
*d*3	4	4	4	4	95	82.1	120	111.2
*d*4	2	2	2	2	90	71.9	110	93.5
*d*5	4	4	4	4	90	88.2	140	108

TABLE 6 Instance summary

Instance	OFV (profit)	Income	Cost	CPU time (s)
E-3	5344.42-optimal	5966	621.58	2101.67

5.1 | Model validation

In this part, the mathematical model is validated through instance E-3. Prices offered by riders for being served are given in Table 4, where $v1$ is the price offered for being served in the morning, $v2$ is the price offered for being served in the evening, and $v3$ is the price offered for being served in both the morning and the evening as a bundle. The cost of traveling per unit of time (h) is 0.68 monetary units. dp means driver number p. For example, $d3$ is driver number 3. mTn means mth rider of type n. For example, $2T3$ is the second rider of type 3.

Additional data and outcome of the test problem are given in Table 5: the capacity of each driver in round 1 (C_1), number of riders served by each driver in round 1 (S_1), capacity of each driver in round 2 (C_2), number of riders served by each driver in round 2 (S_2), maximum time each driver is willing to spend in round 1 (L_1), time spent by each driver in round 1 (T_1), maximum time each driver is willing to spend in round 2 (L_2), and time spent by each driver in round 2 (T_2). The summary of the exact solution outcomes, such as the OFV, are reported in Table 6. The solution of E-3 is illustrated in Figure 7.

5.2 | Exact and heuristic results

Solution results of the test instances are reported in Tables 7 and 8. In Table 7, the exact solution obtained by using CPLEX is also reported; however, in Table 8, there is no exact solution for large-sized problems. In Tables 7, Sol. Type means the type of solution. These tables show the OFV and solution CPU time (Sol. time) in seconds. Also, the mean OFV is computed after five runs. The gap means the gap between the mean OFV of five runs of the heuristic and the exact solution. Best OFV (PSO), Best OFV (DA), Best OFV (ICA), Best OFV (*CA-RS*1), and Best OFV (*CA-RS*2) are the best OFV among five runs of PSO, DA, ICA, *CA-RS*1, and *CA-RS*2, respectively.

The performance of a meta-heuristic algorithm depends on the selection of its parameters that should be tuned carefully. For the sake of brevity, the parameter tuning procedure of each of the meta-heuristic algorithm is detailed in the online Appendix D. Since CA-RS1 and CA-RS2 have the role of improving the solutions provided by the meta-heuristic algorithm, the performance of PSO, DA, and ICA is first assessed and the best one is selected. As can be seen in Tables 7 and 8, all the three meta-heuristic algorithms perform similarly well. To statistically assess the performance of the algorithms and select the best one to be used in the proposed framework, a two-sample t-test is used. First, a two sample t-test is applied to the results of PSO and DA. The t-value obtained is -0.11979 and the p-value is 0.453177. Then, a two-sample t-test method is applied to the results of PSO

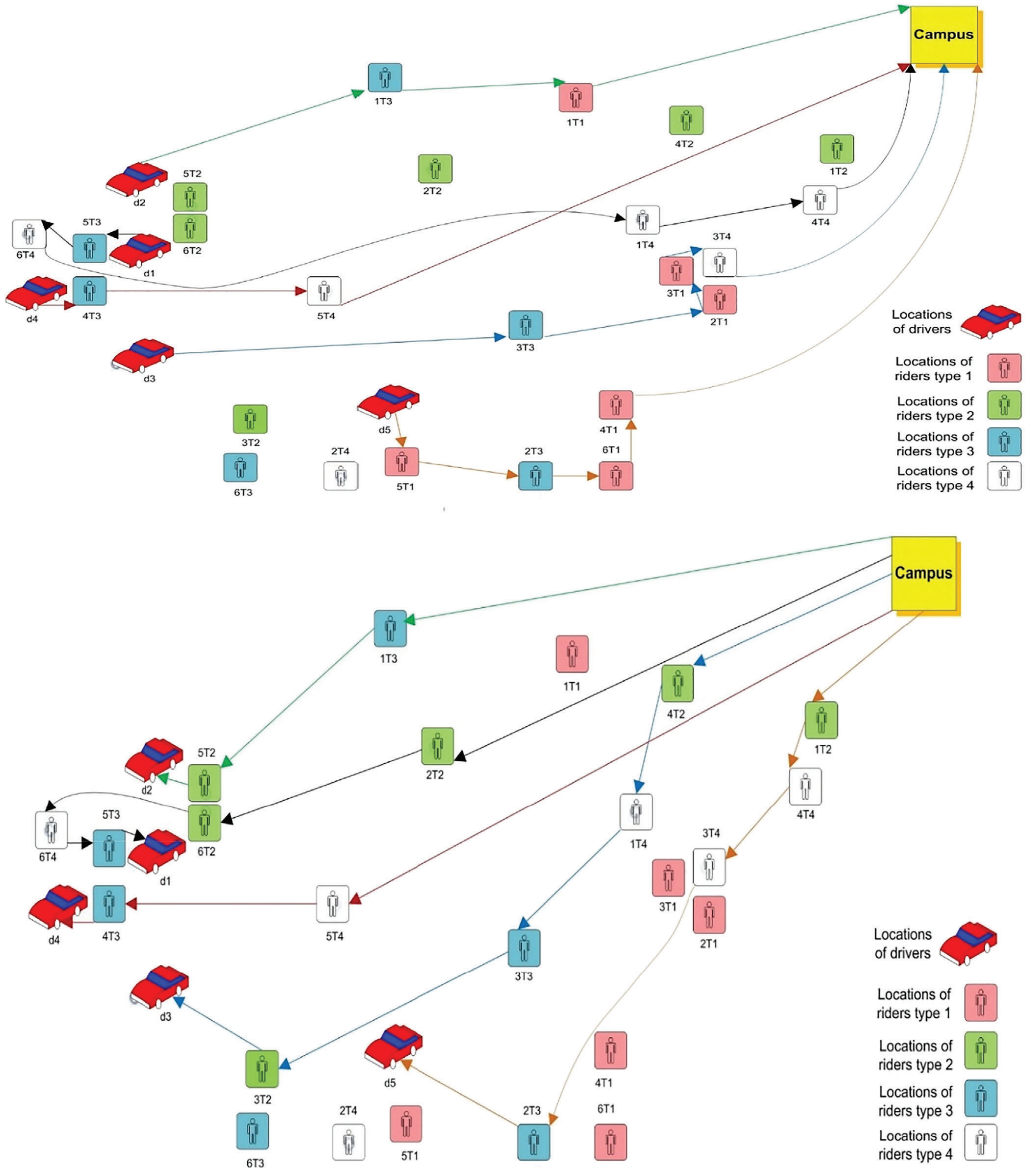


FIGURE 7 Solution of E-3 (inbound and outbound)

and ICA. The *t*-value is −1.00231 and the *p*-value is 0.166602. These tests confirmed that there are no significant differences between our three meta-heuristic algorithms. Therefore, each of them can be selected to be used in our proposed solution framework. In the sequel, we will use PSO. However, there is no difference and any of these three algorithms can be used.

Table 7 shows that the gap between the OFV obtained from the heuristics and the exact solution is always very small, which is evidence that our heuristics framework provides good quality results. This can also be made more evident through a two-sample *t*-test method, which is applied to statistically find the differences between the exact solution value and the CA-RS1 and CA-RS2 solution values. The *t*-value is 0.43725 and the *p*-value is 0.336748 for the exact solution value versus CA-RS1. On the other hand, the *t*-value is 0.42812 and the *p*-value is 0.33993 for exact versus CA-RS1 solutions. These values show

TABLE 7 Solution results (including exact solutions for small problems)

Instance number	Exact solution: Sol. type-OFV- (Sol. Time)	Mean OFV (Mean Sol. Time) Gap		Mean OFV			Best OFV			Best OFV (Sol. Time)	
		*CA-RS*1[a]	*CA-RS*2[a]	PSO	DA	ICA	PSO	DA	ICA	*CA-RS*1[a]	*CA-RS*2[a]
E-1	Optimal 4500.3 (0.312)	4218 (4.6) 6.27%	4233.2 (5) 5.94%	2803.6	2705.9	2911.8	2868.7	2943.8	3120.8	4271.6 (4)	4241.7 (4)
E-2	Optimal 5500 (619)	5306 (17.4) 3.53%	5322.5 (17.2) 3.23%	1783.5	1739	1598.8	1808.8	1752.1	1772.7	5379.1 (20)	5331.4 (17)
E-3	Optimal 5344.4 (2102)	5205.9 (10.2) 2.59%	5196.4 (11.6) 2.77%	1724	1517.8	1619.8	1842.5	1641.1	1638.8	5228.8 (11)	5226.9 (13)
E-4	Optimal 6440.8 (2952)	6149.8 (16.8) 4.52%	6140 (17.6) 4.67%	1130.3	712.2	1162.7	1157.4	762.1	1190.1	6153.5 (17)	6155 (20)
E-5	Feasible[b] 6915.95 (6200)	6538.4 (35.4) 5.46%	6554.4 (45.4) 5.23%	1081.9	1215.4	1109	1139.7	1330.8	1176.5	6609.7 (44)	6620.1 (39)

[a] Since there is no significant difference between the performance of the three proposed meta-heuristic algorithms, in this table and in Table 8, PSO is used to obtain the results of *CA-RS*1 and *CA-RS*2. DA or ICA can be used too, however similar results are expected to be obtained.

[b] In E-5, the solution process was interrupted after 6200 seconds because of a memory fault of CPLEX.

TABLE 8 Solution results (without the exact solution)

Instance number	Mean OFV (Mean Sol. Time)		Mean OFV			Best OFV			Best OFV (Sol. Time)	
	*CA-RS*1[a]	*CA-RS*2[a]	PSO	DA	ICA	PSO	DA	ICA	*CA-RS*1[a]	*CA-RS*2[a]
X-1	10 768.6 (289.2)	10 758.3 (352.4)	3547.1	4163.8	4437.3	3651	4389.44	4595.6	11 123.3 (279)	11 200.2 (347)
X-2	13 000.4 (531)	13 680.2 (730.8)	4101.2	4328.4	3865.7	4392.6	4648.4	4266.3	13 803.1 (473)	13 866.8 (864)
X-3	14 545 (678.8)	14 582.7 (951.4)	5044.6	5137.3	4878.3	5272.6	5460	5411	14 583.1 (709)	14 608 (1033)
X-4	19 306.4 (981)	19 538.2 (1352)	5064.3	5197.2	4816.2	5374.1	5621	4986	19 730.1 (832)	19 764 (1243)
X-5[b]	21 400.4 (1031)	21 568.2 (1630.8)	6043.1	5699.6	5954.5	6312.3	5945.1	6276.8	21 803.1 (973)	22 166.8 (1564)
X-6[b]	22 890.1 (1263)	22 145 (1795)	5716.8	5958	6102.4	6323.5	6447.4	6266.3	22 924 (1325)	22 986.2 (1732)
X-7[b]	23 302.5 (1432)	23 767.1 (1905)	6604.6	6935	6678.7	6872.6	7145.1	7025	23 905.1 (1448)	24 106.9 (1833)
X-8[b]	25 200.4 (1821)	25 368.2 (2024.7)	6498	6171.8	6377	6791	6652.1	6922.1	25 873.8 (1623)	25 196 (2155)
X-9[b]	28 860 (1954)	28 857.3 (2280)	6952.2	6780.7	6884.6	7329.3	7324.2	7133.2	28 896.1 (1892.1)	29 161.4 (2345.6)
X-10[b]	31 191.8 (2157)	31 553.4 (2354.1)	7814.9	7154.1	7676.5	8236.6	7916.4	8266.3	31 931.4 (2015)	32 262.2 (2497.5)

[a] Since there is no significant difference between performance of the three proposed meta-heuristic algorithms, PSO is used to obtain the results of *CA-RS*1 and *CA-RS*2. Similar results should be obtained while employing DA or ICA.

[b] To reduce the solution time while solving these instances by *CA-RS*1 and *CA-RS*2, *Hphase* was implemented on *bpso* and on five randomly selected feasible solutions among *npar* ones at hand after the last iteration of PSO (instead of being implemented on *bpso* and on all *npar* feasible solutions at hand after the last iteration).

that there is no meaningful difference between the exact and *CA-RS*1 and *CA-RS*2 solution values for small-sized instances and prove that the proposed solution framework is statistically reliable and can be used to solve large-sized test instances. Also, Tables 7 and 8 show that almost in all instances, *CA-RS*2 employs longer CPU time with respect to *CA-RS*2, whereas there is not much difference between their OFVs. Thus, we can conclude that *CA-RS*1 has a better performance than *CA-RS*2.

Tables 7 and 8 show that CA-RS1 and CA-RS2 are totally superior to the meta-heuristic algorithms because of the *Hphase*. The *Hphase* was developed based on the presented problem's features to help our framework to improve the provided solutions by meta-heuristics. However, a meta-heuristic algorithm implements a general approach (e.g., PSO simulates birds' behavior) to improve the solution quality of the different problems. Although meta-heuristics these techniques to prevent falling into local optimum solutions, the power of meta-heuristics to improve the solution decreases when the size of problem and its constraints increased. Therefore, they cannot guarantee to find the high-quality solutions, as Table 7 and 8 illustrate.

5.3 | Sensitivity analysis

Now, the performance of the model is going to be assessed from the robustness viewpoint. For this purpose, a sensitivity analysis is made here to investigate the effect of important input parameters on the solution quality.

Problem E-6 is solved with *CA-RS*1, while the cost function (22) is being used, to include the cost (penalty) of riders not served. In Figure 8 and Table 9, OFV is given in terms of different values of the penalty cost g. The parameter g is defined as the cost of not serving a rider at one round (types 1, 3, and 4 riders in the morning and riders of types 2, 3, and 4 in the evening). It is seen that as the value of g increases, OFV decreases. Using the values of 0, 10, 20, and 40 for g, we observe the OFV decreases linearly, and the assignment of riders to drivers does not change. However, at the value of 80 for g, OFV does not appear to change linearly, and this is because the assignment of riders to drivers changes. Then, OFV decreases linearly again when g is 160 and 320, and it means that the assignment of riders to drivers does not change. So, there are two rider–driver assignments, the first one being due to the values of 0, 10, 20, and 40 for g and the second one being due to the values of 80, 160, and 320 for g.

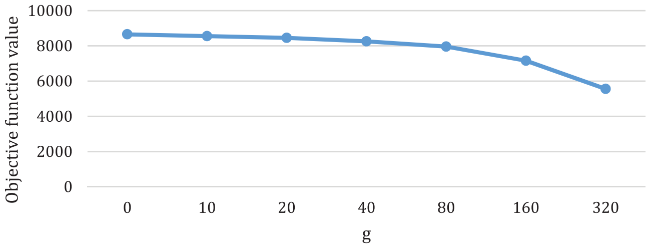


FIGURE 8 OFV for E-6 in terms of *g* (the penalty cost)

TABLE 9 The OFV for E-6 in terms of *g*

g	0	10	20	40	80	160	320
OFV	8654.4	8554.4	8454.4	8254.4	7957.5	7157.5	5557.5

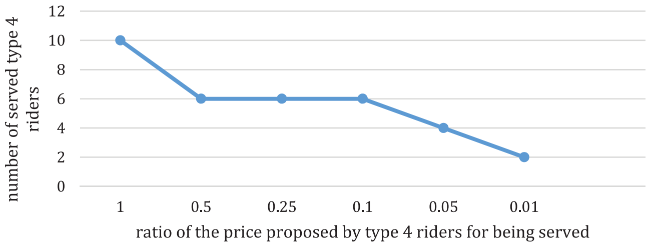


FIGURE 9 Number of served type 4 riders in terms of the proposed prices

Problem E-6 is solved with *CA-RS*1, and the prices offered by type 4 riders for being served are multiplied in a factor [0.01, 1] and gradually reduced. In Figure 9, the number of served type 4 riders is shown in terms of the used factor. It can be seen when the price offered by type 4 riders is reduced, the number of served type 4 riders declines. Given the drivers' constraints, the system is going to serve the riders who offer higher prices. When lower prices are offered by type 4 riders, they will have less chance to be served.

6 | CONCLUSIONS AND FUTURE RESEARCH

Here, a ridesharing system based on the concept of CAs was designed for the students' transportation. This system was formulated using a mixed-integer linear programming model. Then, two variants of a hybrid meta-heuristic-based solution framework were proposed for solving the problem. The model and the solution approach were validated by solving sample problems. To ensure high solutions quality, we proposed a solution framework using three well-known meta-heuristics (i.e., PSO, DA, and ICA), whose performances were assessed. The collected results indicated that there was no significant difference between the forgoing three meta-heuristics. Then, the performance of the two variants of our heuristics was evaluated and the results showed that both variants of the proposed algorithm could find good quality solutions with small gaps compared to the exact solution values. The results also showed that the second variant (CA-RS2) consumed a longer CPU time, without being able to produce better quality solutions compared to the first variant (CA-RS1). Finally, a sensitivity analysis was performed by varying the value of the cost of not serving a rider at any of the rounds, and by varying the price offered by some riders.

Further studies can be developed along with several directions. First, uncertainty can be incorporated in different ways into the proposed model and different auction designs can also be explored, such as multi-round [9], parallel auctions [24], or one-sided uniform prices [5]. The model can also be extended to involve (i) time windows whenever the arrival/departure activities of the students should cover different timeslots of the academic day, (ii) the periodicity aspect of the problem (see [43]), and (iii) alternative and/or multiple objectives and pricing techniques that consider the sustainability, fairness, and service quality aspects of the ridesharing application. Moreover, different heuristic and meta-heuristic algorithms may be explored and compared for solving the defined optimization model as well as for identifying a fair and efficient way of profit sharing and compensation mechanisms among the participants.

ACKNOWLEDGMENT

The first author carried out part of this work while he was affiliated with the Sultan Qaboos University. The publication of this article was funded by the Qatar National Library.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Chefi Triki https://orcid.org/0000-0002-8750-2470
Maziar Mahdavi Amiri https://orcid.org/0000-0003-3812-2348
Reza Tavakkoli-Moghaddam http://orcid.org/0000-0002-6757-926X

REFERENCES

[1] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, *Optimization for dynamic ride-sharing: A review*, Eur. J. Oper. Res. **223**(2) (2012), 295–303.

[2] M. Al-Riyami, N. Ismail, A. Al-Azri, and C. Triki, *A matching optimization model for the dynamic ridesharing in a university campus*, Proceedings of the International Conference on Applied Mechanics and Industrial Systems, 6–8 December, Muscat, Oman, 2016.

[3] A. Amey, *A proposed methodology for estimating rideshare viability within an organization, applied to the MIT community*, TRB Annual Meeting Proceedings, 2011, pp. 1–16.

[4] S. Ben Amor, W. Klibi, and M. Rekik, *A two-stage stochastic model for the winner determination problem in transportation procurement problem*, Proceedings of the ILS2016 (Information Systems, Logistics and Supply Chain), Bordeaux, France, 2016, pp. 1–8.

[5] P. Beraldi, D. Conforti, C. Triki, and A. Violi, *Constrained auction clearing in the Italian electricity market*, Q. J. Belg. French Ital. Oper. Res. Soc. **2**(1) (2004), 35–51.

[6] M. Bruglieri, D. Ciccarelli, A. Colorni, and A. Luè, *PoliUniPool: A carpooling system for universities*, Procedia Soc. Behav. Sci. **20** (2011), 558–567.

[7] C. Caplice and Y. Sheffi, *Combinatorial auctions for truckload transportation*, Combin. Auctions **21** (2006), 539–572.

[8] N. D. Chan and S. A. Shaheen, *Ridesharing in North America: Past, present, and future*, Transp. Rev. **32**(1) (2012), 93–112.

[9] S. De Vries and S. Vohra, *Combinatorial auctions: A survey*, INFORMS J. Comput. **15**(3) (2003), 284–309.

[10] W. Elmaghraby and P. Keskinocak, *Combinatorial auctions in procurement*, in *The Practice of Supply Chain Management*, C. Billington, T. Harrison, H. Lee, and J. Neale, Eds., Kluwer Academic Publishers, Norwell, MA, 2003, 245–258.

[11] S. Erdoğan, C. Cirillo, and J. M. Tremblay, *Ridesharing as a green commute alternative: A campus case study*, Int. J. Sustain. Transp. **9**(5) (2015), 377–388.

[12] P. Fattahi, V. Hajipour, and A. Nobari, *A bi-objective continuous review inventory control model: Pareto-based meta-heuristic algorithms*, Appl. Soft Comput. **32** (2015), 211–223.

[13] M. Furuhata, M. Dessouky, F. Ordóñez, M. E. Brunet, X. Wang, and S. Koenig, *Ridesharing: The state-of-the-art and future directions*, Transp. Res. B: Methodol. **57** (2013), 28–46.

[14] M. Gansterer, R. F. Hartl, and R. Vetschera, *The cost of incentive compatibility in auction-based mechanisms for carrier collaboration*, Networks **73**(4) (2019), 490–514.

[15] M. Guajardo and M. Rönnqvist, *A review on cost allocation methods in collaborative transportation*, Int. Trans. Oper. Res. **23**(3) (2016), 371–392.

[16] S. C. Ho, W. Y. Szeto, Y. H. Kuo, J. M. Leung, M. Petering, and T. W. Tou, *A survey of dial-a-ride problems: Literature review and recent developments*, Transp. Res. B: Methodol. **111** (2018), 395–421.

[17] H. Hosni, J. Naoum-Sawaya, and H. Artail, *The shared-taxi problem: Formulation and solution methods*, Transp. Res. B: Methodol. **70** (2014), 303–318.

[18] L. Hou, D. Li, and D. Zhang, *Ride-matching and routing optimisation: Models and a large neighbourhood search heuristic*, Transp. Res. E: Logist. Transp. Rev. **118** (2018), 143–162.

[19] J. Ignatius, S. M. Hosseini-Motlagh, M. Goh, M. M. Sepehri, A. Mustafa, and A. Rahman, *Multiobjective combinatorial auctions in transportation procurement*, Math. Probl. Eng. **2014** (2014), 1–9.

[20] C. Jiang, J. Fu, and W. Liu, *Research on vehicle routing planning based on adaptive ant colony and particle swarm optimization algorithm*, Int. J. Intell. Transp. Syst. Res. **19** (2020), 83–91.

[21] R. Karamanis, E. Anastasiadis, P. Angeloudis, and M. Stettler, *Assignment and pricing of shared rides in ride-sourcing using combinatorial double auctions*, IEEE Trans. Intell. Transp. Syst. (2020) (in press), https://doi.org/10.1109/TITS.2020.2988356.

[22] K. Kelley, *Casual carpooling enhanced*, J. Public Transp. **10**(4) (2007), 119–130.

[23] C. Triki, *Using combinatorial auctions for the procurement of occasional drivers in the freight transportation: A Case-Study*, J. Clean. Prod. **304** (2021). https://doi.org/10.1016/j.jclepro.2021.127057.

[24] A. Kleiner, B. Nebel, and V. Ziparo, A mechanism for dynamic ride sharing based on parallel auctions, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, Barcelona, Spain, 16–22 July, 2011.

[25] C. Lagos, G. Guerrero, E. Cabrera, A. Moltedo, F. Johnson, and F. Paredes, *An improved particle swarm optimization algorithm for the VRP with simultaneous pickup and delivery and time windows*, IEEE Lat. Am. Trans. **16**(6) (2018), 1732–1740.

[26] A. Y. Lam, *Combinatorial auction-based pricing for multi-tenant autonomous vehicle public transportation system*, IEEE Trans. Intell. Transp. Syst. **17**(3) (2015), 859–869.

[27] J. O. Ledyard, M. Olson, D. Porter, J. A. Swanson, and D. P. Torma, *The first use of a combined-value auction for transportation services*, Interfaces **32**(5) (2002), 4–12.

[28] A. Lee and M. Savelsbergh, *Dynamic ridesharing: Is there a role for dedicated drivers?* Transp. Res. B: Methodol. **81** (2015), 483–497.

[29] Z. Ma, R. H. Kwon, and C. G. Lee, *A stochastic programming winner determination model for truckload procurement under shipment uncertainty*, Transp. Res. E: Logist. Transp. Rev. **46**(1) (2010), 49–60.

[30] R. P. McAfee and J. McMillan, *Auctions and bidding*, J. Econ. Lit. **25**(2) (1987), 699–738.

[31] Y. Meraihi, A. Ramdane-Cherif, D. Acheli, and M. Mahseur, *Dragonfly algorithm: A comprehensive review and applications*, Neural Comput. Appl. **32** (2020), 16625–16646.

[32] M. Mokhtarzadeh, R. Tavakkoli-Moghaddam, C. Triki, and Y. Rahimi, *A hybrid of clustering and meta-heuristic algorithms to solve a p-mobile hub location–allocation problem with the depreciation cost of hub facilities*, Eng. Appl. Artif. Intell. **98** (2021), 104121.

[33] C. Morency, *The ambivalence of ridesharing*, Transportation **34**(2) (2007), 239–253.

[34] A. Mortazavi, A. A. Khamseh, and B. Naderi, *A novel chaotic imperialist competitive algorithm for production and air transportation scheduling problems*, Neural Comput. Appl. **26**(7) (2015), 1709–1723.

[35] M. Nourinejad and M. J. Roorda, *Agent based model for dynamic ridesharing*, Transp. Res. C: Emerg. Technol. **64** (2016), 117–132.

[36] M. Nowak, C. Archetti, and B. Kaminski, *Preface: Special issue on the future of route optimization/vehicle routing*, Networks **73** (2019), 379–381.

[37] T. R. P. Ramos, M. I. Gomes, and A. P. B. Póvoa, *Multi-depot vehicle routing problem: A comparative study of alternative formulations*, Int J Log Res Appl **23**(2) (2020), 103–120.

[38] N. Remli, A. Amrouss, I. El Hallaoui, and M. Rekik, *A robust optimization approach for the winner determination problem with uncertainty on shipment volumes and carriers' capacity*, Transp. Res. B: Methodol. **123** (2019), 127–148.

[39] N. Remli and M. Rekik, *A robust winner determination problem for combinatorial transportation auctions under uncertain shipment volumes*, Transp. Res. C: Emerg. Technol. **35** (2013), 204–217.

[40] D. O. Santos and E. C. Xavier, *Taxi and ride sharing: A dynamic dial-a-ride problem with money as an incentive*, Expert Syst. Appl. **42**(19) (2015), 6728–6737.

[41] M. Stiglic, N. Agatz, M. Savelsbergh, and M. Gradisar, *Enhancing urban mobility: Integrating ride-sharing and public transit*, Comput. Oper. Res. **90** (2018), 12–21.

[42] C. Triki, *Location-based techniques for the synergy approximation in combinatorial transportation auctions*, Optim. Lett. **10**(5) (2016), 1125–1139.

[43] C. Triki and N. Al-Hinai, *Optimization techniques for planning the petrol replenishment to retail stations over a multi-period horizon*, Int. J. Oper. Res. **27**(1–2) (2016), 341–355.

[44] C. Triki, A. Al-Maimani, and J. Akil, *Ridesharing in Muscat: Can it be a sustainable solution for the traffic congestion?* Iran. J. Oper. Res. **8** (2017), 98–109.

[45] C. Triki, S. Mirmohammadsadeghi, and S. Piya, *Heuristic methods for the periodic shipper lane selection problem in transportation auctions*, Comput. Ind. Eng. **106** (2017), 182–191.

[46] C. Triki, S. Piya, and L. L. Fu, *Integrating production scheduling and transportation procurement through combinatorial auctions*, Networks **76**(2) (2020), 147–163.

[47] C. Triki, S. Piya, and L.-L. Fu, *Pre-auction lane selection in an integrated production- distribution planning problem*, Eng. Optim. (2021) (in press), https://doi.org/10.1080/0305215X.2020.1833875.

[48] T. H. N. Vu, A framework for ridesharing recommendation services, *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, Singapore, 2020, pp. 3–11.

[49] X. Wang, M. Dessouky, and F. Ordonez, *A pickup and delivery problem for ridesharing considering congestion*, Transp. Lett. **8**(5) (2016), 259–269.

[50] C. Y. Yan, M. B. Hu, R. Jiang, J. Long, J. Y. Chen, and H. X. Liu, *Stochastic ridesharing user equilibrium in transport networks*, Netw. Spat. Econ. **19**(4) (2019), 1007–1030.

[51] J. J. Q. Yu and A. Y. Lam, *Core-selecting auctions for autonomous vehicle public transportation system*, IEEE Syst. J. **13**(2) (2019), 2046–2056.

[52] J. J. Q. Yu, A. Y. Lam, and Z. Lu, *Double auction-based pricing mechanism for autonomous vehicle public transportation system*, IEEE Trans. Intell. Veh. **3**(2) (2018), 151–162.

[53] B. Zhang, H. Ding, H. Li, W. Wang, and T. Yao, *A sampling-based stochastic winner determination model for truckload service procurement*, Netw. Spat. Econ. **14**(2) (2014), 159–181.

[54] J. Zhang, D. Wen, and S. Zeng, *A discounted trade reduction mechanism for dynamic ridesharing pricing*, IEEE Trans. Intell. Transp. Syst. **17**(6) (2016), 1586–1595.

[55] C. Zhang, J. Xie, F. Wu, X. Gao, and G. Chen, *Pricing and allocation algorithm designs in dynamic ridesharing system*, Theor. Comput. Sci. **803** (2020), 94–104.

[56] B. Zhang, T. Yao, T. L. Friesz, and Y. Sun, *A tractable two-stage robust winner determination model for truckload service procurement via combinatorial auctions*, Transp. Res. B: Methodol. **78** (2015), 16–31.

[57] L. Zheng, P. Cheng, and L. Chen, Auction-based order dispatch and pricing in ridesharing, *2019 IEEE 35th International Conference on Data Engineering*, Macao, 9–11 April, 2019, pp. 1034–1045.

How to cite this article: C. Triki, M. Mahdavi Amiri, R. Tavakkoli-Moghaddam, M. Mokhtarzadeh, and V. Ghezavati, *A combinatorial auction-based approach for ridesharing in a student transportation system*, Networks. **78** (2021), 229–247. https://doi.org/10.1002/net.22074