2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS) | 979-8-3503-2433-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/MASS58611.2023.00051

e-Uber: A Crowdsourcing Platform for Electric Vehicle-based Ride- and Energy-sharing

Ashutosh Timilsina
Department of Computer Science
University of Kentucky
Lexington, USA
ashutosh.timilsina@uky.edu

Simone Silvestri
Department of Computer Science
University of Kentucky
Lexington, USA
simone.silvestri@uky.edu

***Abstract*—There is growing interest in deploying the sharing-economy-based business model to energy systems, with modalities like peer-to-peer (P2P) energy trading, Electric Vehicles (EV)-based Vehicle-to-Grid (V2G), Vehicle-to-Home (V2H), Vehicle-to-Vehicle (V2V), and Battery Swapping Technology (BST). This paper exploits the increasing diffusion of EVs to realize a crowdsourcing platform called *e-Uber* that jointly enables ride-sharing and energy-sharing through V2G and BST. We employ theoretical concepts of *online spatial crowdsourcing*, reinforcement learning, and *reverse auction* to devise this novel platform. Experimental results using real data from New York City taxi trips and energy consumption show that e-Uber performs close to the optimum and outperforms a state-of-the-art approach.**

***Index Terms*—Online spatial crowdsourcing, V2G, energy-sharing, ride-sharing, personalized recommendation, combinatorial multi-armed bandit.**

I. Introduction

With the recent advent of sharing-economy-based models and their successful application in accommodation-sharing (e.g. Airbnb, Vrbo) and ride-sharing (e.g. Uber, Lyft), researchers have focused on applying this concept to energy systems [1]. Energy-sharing modalities such as peer-to-peer (P2P) energy trading [2], [3], and Electric Vehicle (EV)-based Vehicle-to-Grid (V2G), Vehicle-to-Home (V2H), Vehicle-to-Vehicle (V2V) [4], as well as Battery Swapping Technology (BST) [5] have been proposed as sustainable and flexible approaches to balance the energy supply and demand for both the grid and end-users [4], [6]. Especially, the rapid rise in EV sales in recent years has created new opportunities for mobile and flexible energy storage and management including ride-sharing and energy-sharing services using EVs [4]. However, no studies have been made so far to realize a platform that jointly enables both ride-sharing and energy-sharing.

Crowdsourcing is an approach for recruiting workers from a "crowd" to execute tasks [7], [8]. We believe that a crowdsourcing platform has the potential to also be successfully applied to the combined ridesharing and energy-sharing system, where *tasks* are ride- and energy-sharing requests that can be performed by EV drivers, called *workers*. Tasks are requested by *task-requesters* which include ride-sharing clients as well as private or public energy customers. Examples of such energy customers include a utility company and a microgrid community looking to achieve demand response by shifting energy demand to V2G services at different locations, specially during the time of peak energy demands [9]–[12].

In this work, we propose a novel crowdsourcing platform called *e-Uber* that leverages the increasing diffusion of EVs to enable joint ride-sharing and energy-sharing services. With this platform, drivers equipped with EVs can not only transport passengers through ride-sharing but also sell excess energy stored in their batteries to the grid/houses during periods of high demand through V2G or battery swapping [13]–[15]. e-Uber has the potential to increase the earning potential for drivers and also to help balance the energy demand and supply for the grid while simultaneously fulfilling the mobility and energy demands of consumers.

A few works on crowdsourcing have been proposed to facilitate the integration of energy-sharing services with EVs. Ai et al. [6] proposed a V2H-based omni-sharing modality in a microgrid community to crowdsource energy from EVs. Similarly, the authors in [16] propose an autonomous EV (AEV)-based energy crowdsourcing approach, allowing AEVs to participate in energy-sharing tasks for consumers placed in the cloudlet. However, these approaches do not consider the *workers' preferences* as well as their *limited ability* of selecting tasks when overwhelmed with choices and problems. There have been a few spatial crowdsourcing work attempting at solving the task assignment problem considering worker preferences [8], [17], [18]. However, these approaches focus on general uniform tasks, and do not consider ride-sharing combined with energy-sharing.

To the best of our knowledge, in this paper we propose the first crowdsourcing mechanism that jointly enables ride- and energy-sharing to provide a multifaceted solution to existing problems on efficiency and sustainability of transportation, energy management, and cost-effective demand response using EVs. *e-Uber* works in three decision stages: calculate a personalized task recommendation for each EV worker, collect bids from workers, reverse auction-based winning bids selection. We propose a preference-aware optimal task recommendation problem, $POTR$, and a reinforcement learning mechanism, called $CARS$ to solve this $POTR$ problem by learning the workers' preferences over time based on their interaction with the recommendations using Combinatorial Multi-Armed Bandit framework [1]. Reverse auction is used for bidding and the

2155-6814/23/$31.00 ©2023 IEEE
DOI 10.1109/MASS58611.2023.00051

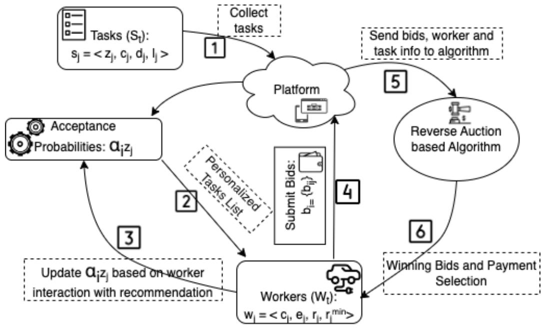

Fig. 1. Working mechanism of e-Uber

winning bids are determined through an optimization framework called *Winning Bid Selection* ($WiBS$). Proving that $WiBS$ problem is NP-hard, we propose bipartite matching-based heuristic, BMW that finds solution to $WiBS$ in polynomial time. Through extensive experiments using real data, we show that *e-Uber* outperforms state-of-the-art solutions and realizes crowdsourcing of energy and ride-sharing. Extended version of this paper can be found in the arXiv [19].

II. System Model

We assume time to be divided in time slots. At each time slot t, the set of tasks is referred to as $\mathcal{S}_t$, which are crowdsourced to the workers. We refer to $\mathcal{W}_t$ as the set of workers at time t. Each task in $\mathcal{S}_t$ is denoted by a tuple $s_j \stackrel{\text{def}}{=} \langle z_j, c_{s_j}, d_j \rangle$ where z_j is the type of task ($0-$rideshare, $1-$battery swapping, and $2-$V2G), c_{s_j} is the start position and d_j is the destination of task. For V2G tasks, although spatial in nature, start position c_{s_j} is same as destination d_j. We assume the utility company submits V2G tasks as a result of an *energy requirement* $\mathcal{E}$. This is a typical assumption for demand response solutions [9]–[11]. As a result, the total amount of energy provided by workers through V2G must be at least $\mathcal{E}$. Each worker in $\mathcal{W}_t$ is denoted by a tuple $w_i \stackrel{\text{def}}{=} \langle c_{w_i}, e_i, r_i, r_i^{min} \rangle$, where c_i is the current position of the EV worker w_i which can be different to spatial task location c_{s_j}, e_i is the energy per unit range value in (kWh/km) that gives information about how much energy the EV consumes to drive a unit distance, r_i is the available range of electrical vehicle in km given by the remaining energy level in their batteries, and r_i^{min} is the minimum energy not to be exceeded after completing the task to ensure sufficient energy for traveling to a charging location. The energy required to perform task s_j by worker w_i is denoted by l_{ij}. e-Uber provides a list of tasks, called *recommendation list*, to each worker. Workers then submit bids to these tasks. The bid $b_{ij} \in \mathcal{B}$ represents the cost asked by worker w_i to perform task s_j, where $\mathcal{B}$ is the set of all the bids submitted by workers.

Existing works in crowdsourcing and energy-sharing using EVs have generally assumed that workers would have complete access to the list of available tasks and would pick the best task for them or, conversely, the crowsourcing platform would assign tasks to workers regardless of their preference. These assumptions are both undesirable. On the one hand workers have limited time and ability to go over potentially a very long list of tasks [1], and on the other, workers may have different preferences to different tasks. In this work, we recommend a limited list of relevant tasks to each worker based on their preferences. We model the preferences as $\alpha_{iz_j} \in [0, 1]$ denoting the probability that worker w_i bids for a task of type z_j. These are called *bidding probabilities*. We assume that these probabilities are unknown and thus need to be learned over time by observing the workers' behavior.

III. *e-Uber:* Problem Formulation

Fig. 1 summarizes the steps involved in the e-Uber platform. e-Uber collects a list of tasks $\mathcal{S}_t$ at time t as requested by task-requesters which need to be crowdsourced to the EV-based workers in $\mathcal{W}_t$ (step 1). The platform sends a personalized list of tasks to the workers based on their preferences (step 2) to which they respond by submitting bids to the platform for the tasks (step 3). Based on the received bids $\mathcal{B}_t$ (step 4), e-Uber uses reverse auction based algorithm to determine the winning bids $\mathbf{q}^*$ and final payment $\mathbf{P}$ for winners (step 5). Finally, the worker preferences are updated based on their feedback for the next time step (step 6). Given the nature of the considered tasks, worker-task assignment is performed one-to-one.

As described above, the system involves solving two different problems. One is to recommend a set of tasks which maximizes the likelihood of generating the maximum number of bids, and thus improving the overall system performance. Another problem is to select the winning bids for task assignment and determine the final payment to crowdsource the tasks to the workers. These two problems are discussed below.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{w_i \in \mathcal{W}} \sum_{s_j \in \mathcal{S}} \alpha_{iz_j} x_{ij} && && (1)\\
\text{s.t.} \quad & \sum_{s_j \in \mathcal{S}} x_{ij} \leq K, && \forall w_i && (1a)\\
& \sum_{w_i \in \mathcal{W}} x_{ij} \geq \psi, && \forall s_j && (1b)\\
& \sum_{s_j \in \mathcal{S}} g(z_j) x_{ij} \geq \frac{|V2G|}{|\mathcal{S}|} K, && \forall w_i && (1c)\\
& l_{ij} x_{ij} \leq (r_i - r_i^{min}) e_i, && \forall w_i, s_j && (1d)\\
& x_{ij} = 0, \text{ if } |c_{s_j} - c_{w_i}| > \lambda, && \forall w_i, s_j && (1e)\\
& x_{ij} \in \{0, 1\}, && \forall w_i, s_j && (1f)
\end{aligned}
$$

A. Preference-aware Optimal Task Recommendation Problem

Our objective is to recommend a limited subset of tasks to each workers which maximizes the likelihood of bidding for these tasks, while avoiding to overwhelm workers with a list above their cognitive capabilities. We formalize this through the Preference-aware Optimal Task Recommendation (POTR) problem as follows. In short, the problem aims at maximizing the overall task bidding probabilities (hereafter referred interchangeably as preferences) while limiting the size of the recommended list to K as well as ensuring that each task is recommended to at least ψ workers.

The objective function in Eq. (1) maximizes the sum of individual bidding probabilities for each worker's recommended tasks. The binary decision variable $x_{ij} \in \{0,1\}$ is set to 1 if the task s_j is included in the list of worker w_i. Constraint (1a) limits the length of each recommendation list to K. In constraint (1b), each task is recommended to at least $\psi = \left\lfloor \frac{|\mathcal{W}|K}{|\mathcal{S}|} \right\rfloor$ workers. Also, we ensure that a minimum of $\frac{|V2G| \times K}{|\mathcal{S}|}$ V2G tasks are recommended to each workers in constraint (1c). Constraint (1d) limits the energy consumption for each recommended task, ensuring that an EV has enough energy after performing that task to drive to charging location, if required. Finally, constraint (1e) ensures that only the tasks within λ distance from workers are recommended.

The information on bidding probabilities is difficult to obtain *a priori* as it is specific for each worker and include elements of complex human psychology. Therefore, we assume that the preferences are initially unknown and are learned by observing the workers' behavior with respect to the assigned tasks. In section IV, we present a *Combinatorial Multi-Armed Bandit (CMAB)*-based approach [1] to learn such preferences.

B. Winning Bid Selection and Final Payment Problem

After sending the personalized list of tasks to each worker, *e-Uber* collects the bids. Given the collected bids, e-Uber selects winning bids, i.e., the workers performing the tasks, by solving the Winning Bid Selection ($WiBS$) problem. This problem determines the best bids which minimize the total cost from perspective of task requesters. $WiBS$ can then be formulated a costrained assignment problem as follows:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{w_i \in \mathcal{W}} \sum_{s_j \in \mathcal{S}} b_{ij} q_{ij} && (2)\\
\text{s.t.} \quad & \sum_{s_j \in \mathcal{S}} q_{ij} \leq 1, && \forall w_i \quad (2a)\\
& \sum_{w_i \in \mathcal{W}} q_{ij} = 1, && \forall s_j, z_j < 2 \quad (2b)\\
& \sum_{w_i \in \mathcal{W}} q_{ij} \leq 1, && \forall s_j, z_j = 2 \quad (2c)\\
& \sum_{w_i \in \mathcal{W}} \sum_{s_j \in \mathcal{S}} g(z_j) l_{ij} q_{ij} \geq \mathcal{E}, && (2d)\\
& q_{ij} \in \{0,1\}, && \forall w_i, s_j \quad (2e)
\end{aligned}
$$

The objective function in Eq. (2) minimizes the total cost of performing tasks from the collected bids. q_{ij} is the binary decision variable as defined in constraint (2e) that indicates whether a bid b_{ij} wins the auction and therefore the task s_j is assigned to worker w_i. Constraint (2a) ensures that a worker is assigned at most one task, while (2b) allows a ride-sharing and battery swapping tasks ($z_j < 2$) to be assigned to only one worker. Similarly, constraint (2c) ensures that a V2G task is assigned to at most one worker. Finally, constraint (2d), ensures that at least $\mathcal{E}$ amount of energy will be supplied through V2G services. Note that the function $g(z_j) = 1$ if $z_j = 2$ (V2G task) and zero otherwise.

Following the winning bids selection by solving the $WiBS$ problem in Eq. (2), the final payment for each winning worker w_k assigned with task s_j is the second-to-the-selected bid received for that task. Since with the second price payment rule, the dominant strategy for all bidders is to bid truthful [20], it ensures rational workers will provide truthful bids.

Theorem 1. *$WiBS$ problem defined in Eq.* (2) *is NP-hard.*

Proof. The theorem can be proven providing a reduction from the 0-1 min Knapsack (0-1 min-KP) problem [21]. The proof is omitted due to space limitations (refer [19] for proof). □

IV. E-UBER SOLUTION APPROACHES

A. CMAB-based Task Recommendation System

In order to solve the optimization problem in Eq. (1), it is necessary to have beforehand knowledge on the workers preferences. These are generally *not known* a priori in realistic settings. Therefore, it becomes necessary to learn these preferences during run-time, while simultaneously optimizing the task assignment. To this purpose, we propose a reinforcement learning approach inspired by the Combinatorial Multi-Armed Bandit (CMAB) framework [1], [22].

Combinatorial Multi-Armed Bandit is a classic reinforcement learning problem that consists of setup where agents can choose a combination of different choices (i.e. certain decision-making *actions*) and observe a combination of linear *rewards* at each timestep. The long term objective for the problem is to find a strategy that maximizes such reward by selecting optimal actions. This strategy, better defined as *policy*, needs to be learned based on how the agents choose to interact with the system. The learning is carried out through *exploration vs. exploitation* trade-off. Since, at the beginning, the knowledge about how an agent chooses to engage with the system is not known, the system learns by allowing agent to choose from diverse options and therefore learning the user interaction accordingly, referred to as *exploration*. With time, the system gathers information about agent's behavior and uses that knowledge instead of sending out diverse range of choices, called *exploitation*. By balancing this exploration and exploitation mechanism over the course of time, the system eventually picks up on agent's behavior and learns optimal strategy for them. In our problem setting, the workers are the agents who needs to be sent out an optimal set of tasks so as to accumulate good quality bids from them. Specifically, the objective is to find the best possible task recommendations (actions) to be sent to each workers (agent) that will result in higher cumulative preferences for workers (reward).

Hence, in this section, based on this CMAB framework, we design an algorithm called *CMAB-based Algorithm for task Recommendation System (CARS)*. *CARS* recommends the personalized tasks to each workers based on current estimation of worker preferences towards each task type. Note that the worker preference is defined as the bidding probability in section III that a worker will submit a bid for any task based on its type. The algorithm then updates and learns these bidding probabilities based on the worker's engagement on the recommendation through bids. If the worker submits a bid, it is considered to be a preferred recommendation and opposite, if the worker chooses to ignore by not the submitting bid. Based on this information, the preference of workers towards each task type is updated.

Therefore, with $\mathcal{F}$ as the overall solution space that consists of all feasible action matrices, the action matrix $\mathbf{A}(t) \in \mathcal{F}$ corresponds to the optimal set of recommendation lists for the timestep t. It consists of action values $x_{ij} \in \{0,1\}$, which is same as the decision variable in POTR problem. Recall that it represents whether the task s_j is in personalized recommendation list of worker w_i for timestep t. Given this action matrix, the preference of worker w_i towards each task type z_j is modeled as a random variable $\bar{\alpha}_{iz_j}$ whose mean value is α_{iz_j} and is initially unknown. The current knowledge until timestep t for these random variables $\bar{\alpha}_{iz_j}$ is denoted by the estimated expected $\widehat{\alpha}_{iz_j}$. The reward for the platform for selecting the action matrix $\mathbf{A}(t)$ at timestep t, is defined as the sum of the preferences to each workers:

$$\mathbf{R}_{\mathbf{A}(t)}(t) = \sum_{w_i, s_j} x_{ij}(t)\bar{\alpha}_{ij}(t) \tag{3}$$

Since the distribution of $\bar{\alpha}_{iz_j}$ is unknown, the goal of this CMAB-based approach is to learn the policy, that minimizes the overall *regret* up to time t. This regret is defined as the difference between expected reward with perfect knowledge of preferences and that obtained by the policy over time:

$$\mathcal{R}(t) = t\mathbf{R}^*_{\mathbf{A}(t)}(t) - \mathbb{E}\Big[\sum_{t'=1}^{t} \mathbf{R}_{\mathbf{A}(t')}(t')\Big], \tag{4}$$

where $\mathbf{R}^*_{\mathbf{A}(t)}(t)$ is the optimal reward obtained with perfect knowledge of the preference variables. Even though minimizing the regret is a difficult problem, $CARS$ ensures that the regret is bounded, meaning the non-optimal actions will be picked only a limited number of times and eventually the learned policy will converge towards optimal. We present a modified objective function from UCB1 algorithm to select the action matrix as follows.

$$\mathbf{A}(t) = \arg\max_{\mathbf{A}\in\mathcal{F}} \sum_{w_i\in\mathcal{W}} \sum_{s_j\in\mathcal{S}} x_{ij} \left(\widehat{\alpha}_{iz_j} + \sqrt{\frac{(Q+1)\ln t}{m_{iz_j}}} \right) \tag{5}$$

where $Q = |\mathcal{W}| \times |z_j|$ is the total number of variables and m_{iz_j} is the number of observations so far for the variable $\bar{\alpha}_{iz_j}$.

At each timestep t, we solve the $POTR$ problem with CMAB-based objective function in Eq. (5) instead of Eq. (1) and same constraints (1a)-(1f). By solving this modified problem, the sets of optimal actions (or recommendation lists) for each workers are selected based on current estimate of preferences until timestep $(t-1)$. For this purpose, we keep track of the $\widehat{\alpha}_{iz_j}$, along with m_{iz_j}. These two information are then used to update the current estimation of the variable $\bar{\alpha}_{iz_j}$ at time t based on the worker's engagement with the recommendation i.e. whether the worker chooses to submit the bid or not. Needs to be noted that, if the worker chooses to submit the bid, they must complete the task if assigned.

$$\widehat{\alpha}_{iz_j}(t) = \begin{cases} \frac{\widehat{\alpha}_{iz_j}(t-1)m_{iz_j}(t-1)+\alpha_{iz_j}(t)}{m_{iz_j}(t-1)+1} & \text{if } 0 < b_{ij} < \infty, \\ \widehat{\alpha}_{iz_j}(t-1) & \text{otherwise.} \end{cases} \tag{6}$$

$$m_{iz_j}(t) = m_{iz_j}(t-1) + 1 \tag{7}$$

However, as shown in Theorem (1), finding optimal solution for winner determination problem ($WiBS$ problem Eqs. (1)-(2e)) is NP-Hard problem. Therefore, we devise a bipartite matching-based heuristic for winning bid determination with polynomial time complexity for worker-task assignment.

B. Winning Bid Selection using Weighted Bipartite Matching

The $WiBS$ problem formulation in Eq. (2) is an extension of one-to-one weighted matching. However, this matching has to select minimum weighted edges for task allocation with energy budget constraints for V2G tasks. Therefore, we hereby develop a heuristic inspired by bipartite minimum weighted matching which can be solved in polynomial time using Karp's algorithm [23]. To satisfy the energy budget constraint, we employ iterative matching that removes the highest weighted edges from the previous matching until the budget is met. Simply put, the algorithm runs the minimum weighted matching and if it does not satisfy the budget constraints, removes first z highest weighted edges connected to non-V2G tasks from the previous matching and then runs another round of matching until the feasible solution is found.

This algorithm called *Bipartite Matching-based Winner selection (BMW)* is presented in Alg. 1. BMW takes set of available workers $\mathcal{W}$, tasks $\mathcal{S}$, and the set of bids $\mathcal{B}$ as input and finds the winning bids with final pay P as the output. In line 1, the algorithm initializes the output graph Φ_{out}, a temporary graph Φ_{temp} for iterative matching purpose, and P. Then it creates a separate sets for V2G and non-V2G tasks as sets V and R in line 2 and collects the bids from all workers (line 3). With the information on bids, BMW generates a bipartite graph G between bipartite sets of workers $\mathcal{W}$ and tasks $\mathcal{S}$, and adds edges between those nodes that have non-zero bids i.e. worker w_i with non-zero bid b_{ij} is connected with task s_j (lines $4-7$). Now, it runs a bipartite matching iteratively with while loop in lines $8-15$. Initially, both of the conditions for while loop are true and therefore the algorithm runs first round of Minimum Weighted Bipartite Matching on graph G (line 9). It then assigns the matched graph to the output graph Φ_{out} (line 10) and checks if the energy budget for V2G tasks is satisfied (line 11). If it is met in the first round, it breaks out of the while loop and determines final payment and task assignment. If it is not met, BMW removes the first z highest weighted edges in Φ_{out} from G that just meet the remaining of energy budget not met (line $12-13$). Then, since both of the conditions are still true, the algorithm runs another round of matching on reduced graph G. Eventually the

Algorithm 1: <u>B</u>ipartite <u>M</u>atching-based <u>W</u>inner selection (BMW)

Input : Sets of Workers ($\mathcal{W}$) and Spatial Tasks ($\mathcal{S}$), Bids ($\mathcal{B}$)
Output: Winning bids with final pay ($\mathbf{P}$)
/* **Initialization** */
1 $\Phi_{out} = \{\mathcal{W} \cup \mathcal{S}, E_{\Phi} = \emptyset\}; \Phi_{temp} = \emptyset; P = \emptyset$;
/* **Generate bipartite graph** G */
2 $\forall s_j \in \mathcal{S}$, **if** $g(z_j) = 1$ **then** $V \leftarrow \{s_j\}$ **else** $R \leftarrow \{s_j\}$;
3 $\forall w_i \in \mathcal{W}$, collect their respective bids $\mathcal{B}_i$;
4 Build Bipartite Graph $G = \{\mathcal{W} \cup \mathcal{S}, E_G = \emptyset\}$;
5 **for** *each* $w_i \in \mathcal{W}, s_j \in \mathcal{S}$ **do**
6   **if** $b_{ij} > 0$ **then** Add edge (w_i, s_j) to E_G with weight, b_{ij};
7 **end**
/* **Run minimum bipartite matching until termination** */
8 **while** $\sum_{(w_i,s_j)\in E_{out}} g(z_j) l_{ij} < \mathcal{E}$ *or* $\Phi_{temp} \neq \Phi_{out}$ **do**
9     $E_{out} \leftarrow$Perform Minimum Weighted Bipartite Matching on G;
10   Output graph $\Phi_{out} = \{\mathcal{W} \cup \mathcal{S}, E_{out}\}$, where $E_{out} \subseteq E_G$;
    /* Remove edges if V2G energy budget is not met, and run MWBM on reduced G again */
11   **if** $\sum_{(w_i,s_j)\in E_{out}} g(z_j) l_{ij} < \mathcal{E}$ **then**
12         $Z \leftarrow$Select the first z highest weight edges $\in \Phi_{out}$ and R s.t. $\left(\sum_{(w_i,s_j)\in E_{out}} g(z_j) l_{ij} + \sum_{(w_i,s_j)\in Z} l_{ij}\right) \geq \mathcal{E}$;
13   **if** $Z \neq \emptyset$ **then** Remove all edges $\in Z$ from G and Φ_{out} **else** $\Phi_{temp} = \Phi_{out}$;
14   **end**
15 **end**
16 $\mathbf{q}^* = E_{out}$;
/* **Final Payment and Task Assignment** */
17 $\forall w_k \in \mathcal{W}, P_k \leftarrow$ Second to the selected bid b_{kj};
18 Assign the tasks to winning workers along with final price $\mathbf{P}$;

final matching in output graph Φ_{out} is used as winning task assignments with final payment as per the bid (line $16-18$).

V. Experimental Results

A. Experimental Setup

Our experimental setup consists of modeling workers, tasks and the simulation platform. In case of workers, we gathered the publicly available data on 54 different EV models on battery size, range, charging power and charging speed, and formulated an individual profile for each EV in concern. Similarly for ride-sharing tasks, the high volume taxi trip data of New York City (NYC) from the year of 2013 [24] was used. The V2G tasks were generated from the 15 minutes energy consumption data from 25 NYC residences from PecanStreet [25]. In absence of real dataset on battery swapping tasks, half of the ride-sharing tasks were extracted as the battery swapping tasks, given their similar profile with batteries transported instead of passengers. These tasks are spatial, therefore, we collect the information on locations, distance, and time required to complete the tasks. Bids were generated using a Deep Neural Network (DNN) model. We used 11 months of taxi data to train the DNN model with 80-20 train-test split. The DNN model consisted of 3 hidden layers of sizes $(132, 132, 64)$. We employed ReLU activation function as well as one-hot encoding for the input features.

Furthermore, the simulation platform, e-Uber is developed in Python using Gurobi, NetworkX, and PyTorch libraries. We consider a reverse auction period resolution of 15 minutes which corresponds to the resolution set by grid for real-time energy trading. This means that every 15 minutes the e-Uber algorithm will collect the tasks, push the personalized list of tasks to workers, collect the bids and assign the tasks to EV workers that minimizes the overall cost for the task requesters. We set the search radius for the tasks $\lambda = 10$ km and the maximum length of recommendation list $K = 5$. The energy budget for each 15 minutes time period was considered to be total of all 25 V2G tasks available. The user preferences were sampled uniformly from the set $\{0.1, 0.4, 0.5, 0.7, 0.9, 1.0\}$. The energy, time and location of the EVs are tracked and updated accordingly so as to simulate their real-world trip behavior. If the battery level of the cars fall below a minimum, they are considered for the charging for the next time-step.

For comparison approach, we use the task-centric winner selection algorithm from [26] and refer it as BG for baseline greedy. This approach neither considers user-preference in the problem-setting nor it considers the personalized recommendation system. So for fair comparison, we augment this method with perfect knowledge-based recommendation system that pushes K best tasks as recommendation to each workers. Then we implement the algorithm as presented in [26] that sorts the bids from lowest to highest for each tasks and assigns them one by one. Note that this approach may not guarantee a complete matching between workers and tasks as the tasks that are processed towards the end may not have any workers left to choose from because of limited number of bids and greedy selection approach. We use this BG as our baseline and compare the performance of our algorithms $CARS$ and BMW along with their perfect knowledge variation PK which has the perfect knowledge on the worker preferences and thus do not involve learning, and OPT optimal solution to $WiBS$ problem. The ride-share dataset in concern consists of actual ride-fare for specific car. However, we require bids from each vehicle for recommended tasks and a realistic model for bid generation is quite difficult to obtain. Therefore we trained a Deep Neural Network with existing dataset for determining the ride fare of the given ride-sharing tasks, the details of which is presented in the following.

B. Results

1. <u>**Performance over time – Total Cost and # of Tasks**</u>: In the first experimental scenario, we observe the performance of algorithms as a snapshot of objective values over 24 hours (i.e. $24 \times 4 = 96$ timeslots). We present the objective values from midnight to next midnight as a lineplot in Fig. 2 and cumulative bar plots of objective values (Fig. 3) and total tasks completed (Fig. 4) over a day. Although all the proposed approaches start from the same initial state (except for knowledge on preference), these algorithms may have different successive states since the solution is affected by the matching in previous timeslot, availability of specific workers for next round, and the distance travelled by these workers for previous assignment (or next assignment). Therefore, we employ cumulative objective values and cumulative tasks completed as the metric for a fair comparison of the approaches in Fig. 3.

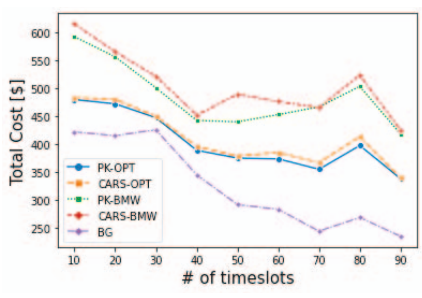

Fig. 2. Snapshot of total cost vs. time

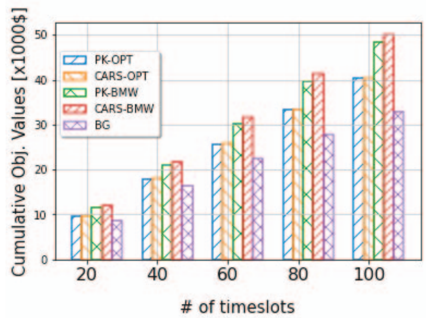

Fig. 3. Cumulative obj. values

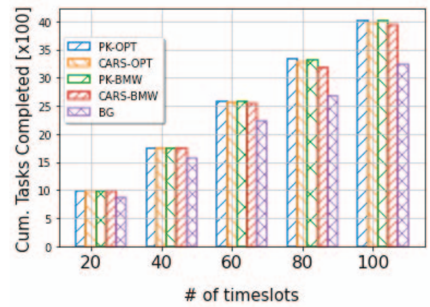

Fig. 4. Cumulative tasks

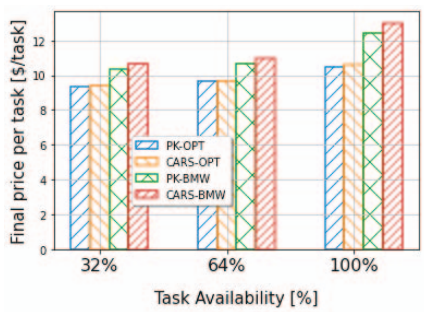

Fig. 5. Avg. Price/task vs. Task(%)

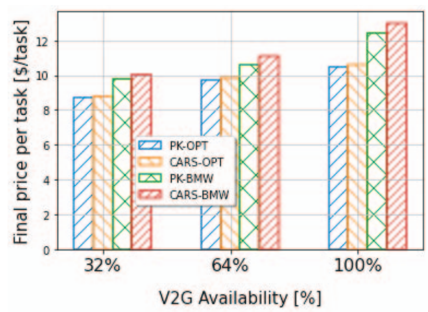

Fig. 6. Avg. price/task vs. V2G (%)

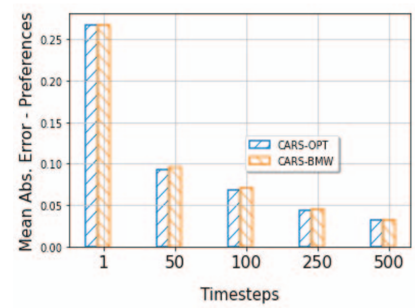

Fig. 7. Mean Absolute Error vs. time

This cumulative objective value reflects the overall quality of task assignment made so far based on the total objective values to achieve the requirement while the cumulative tasks completed present the total number of matches made by the respective approach until the end of that timeslot. As seen in the lineplot Fig. 2 and barplot Fig. 3, the solution generated by baseline greedy approach BG is the minimum one as it assigns task based on respective cheapest bid available but it doesn't meet the maximum number of matching possible unlike other approaches as shown in Fig.4. Therefore, BG mostly violates the V2G requirement, meaning it generates infeasible solutions and hence fails for this problem setting. The $PK - OPT$ produces the best result since it involves solving the $POTR$ and $WiBS$ problem optimally with perfect knowledge of the worker preferences. Following it, is the optimal solution OPT paired with our proposed learning framework for e-Uber, $CARS$, which performs close to optimal in terms of both objective values and number of tasks completed. Although this approach $CARS - OPT$ finds optimal solution, it does not have initial knowledge on preferences. Therefore, it generates sub-optimal recommendation list which then affects the solution to $WiBS$ problem and hence, the overall performance. However, even with online learning framework employed, it produces similar results to the $PK - OPT$.

Also we observe similar pattern with $PK - BMW$ and $CARS - BMW$ since they both rely on bipartite matching-based approach to find feasible solution. Since $PK - BMW$ sends the optimal recommendation to workers for collecting bids, it therefore has higher overall performance compared to $CARS - BMW$ which learns the preferences over time. The price gap between best performing $PK - OPT$ and worst performing $CARS - BMW$ however is less than $\$150$ in the worst case which amounts to a price hike of $\sim \$3/$task with an average 50 tasks for a timeslot. We observe the cumulative objective values grow almost linearly for all approaches and as expected, the performance observed was better for $PK - OPT$ followed by $CARS - OPT$ and then $PK - BMW$ and finally $CARS - BMW$. However, the gap in cumulative objective value increased for the bipartite heuristic compared to optimal due to its sub-optimal performance. Note that the baseline BG generates less cumulative objective value but it fails to generate maximal matching as seen in Fig. 4. The number of tasks completed by the proposed approaches exceed 850 more than the BG in the span 24 hours.

2. **Average final price per task and scaling**: In this experiment, we track the average final price per task while scaling the available tasks from 32% to 64% and then at 100%. For scaling the tasks, we increase the number of each type of tasks proportionally. The result is plotted in Fig. 5. As the system scales, the average final price per task for all approaches rises since the overall cost for the system also increases with the tasks. However, it is also observed that $CARS - BMW$ and $BMW - PK$ suffer more as we scale the system. The margin between these and optimal approaches grows drastically up to $\sim$ $\$2$. This can be attributed mainly to the increased complexity of the problem as number of tasks is increased and

hence the bipartite matching-based heuristic finds less efficient solution compared to optimal. The optimal solutions however have nominal increase in their average price per task ($\sim \$10$) even with scaling compared to rest. We also study the effect of scaling V2G tasks to the average final price per task in Fig. 6. We observed similar trend to above but with noticeable gap between optimal and heuristic approaches when only 32% of V2G tasks are available. This results from the sub-optimal performance owing to less number of V2G tasks compared to rest and hence unequal rate of learning the preferences.

3. **Learning accuracy for preferences – MAE**: To study the learning accuracy of proposed $CARS$ algorithm in conjunction with optimal and BMW, we use the Mean Absolute Error (MAE) of the learned preferences over time as in Fig. 7. Both approaches use the same learning algorithm but the solution to $WiBS$ problem differs and thus affects the learning performance. However, this difference is very negligible. Initially, the MAE is 0.28, which rapidly decreases to less than 0.05 for both approaches by 250 timesteps, showing marginal difference in learning efficacy between them. Since by 500 timesteps the system has garnered sufficient knowledge on workers preferences, MAE falls to 0.03 reflecting the efficacy of proposed CMAB-based preference learning.

VI. Conclusion

e-Uber is a promising crowdsourcing platform for improving the efficiency and sustainability of ride-sharing and energy-sharing services through the use of EVs. It uses reverse auction mechanism to assign spatial tasks to EV drivers based on their preferences, battery level, and other realistic constraints like minimum energy requirement for grid and one-to-one assignment. Results using real data show that e-Uber outperforms recently proposed state-of-the-art solutions.

Acknowledgment

This work is supported by the NSF grant EPCN-1936131 and NSF CAREER grant CPS-1943035.

References

[1] A. Timilsina, A. R. Khamesi, V. Agate, and S. Silvestri, "A reinforcement learning approach for user preference-aware energy sharing systems," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1138–1153, 2021.

[2] A. Timilsina and S. Silvestri, "Prospect theory-inspired automated p2p energy trading with q-learning-based dynamic pricing," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 4836–4841.

[3] W. Tushar, T. K. Saha, C. Yuen, D. Smith, and H. V. Poor, "Peer-to-peer trading in electricity networks: an overview," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3185–3200, 2020.

[4] M. Shurrab, S. Singh, H. Otrok, R. Mizouni, V. Khadkikar, and H. Zeineldin, "An efficient vehicle-to-vehicle (v2v) energy sharing framework," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5315–5328, 2021.

[5] M. R. Sarker, H. Pandžić, and M. A. Ortega-Vazquez, "Optimal operation and services scheduling for an electric vehicle battery swapping station," *IEEE transactions on power systems*, vol. 30, no. 2, pp. 901–910, 2014.

[6] W. Ai, T. Deng, and W. Qi, "Crowdsourcing electrified mobility for omni-sharing distributed energy resources," in *AI and Analytics for Smart Cities and Service Systems*. Springer, 2021, pp. 365–382.

[7] F. Restuccia, P. Ferraro, S. Silvestri, S. K. Das, and G. L. Re, "Incentme: Effective mechanism design to stimulate crowdsensing participants with uncertain mobility," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1571–1584, 2018.

[8] F. Restuccia, P. Ferraro, T. S. Sanders, S. Silvestri, S. K. Das, and G. L. Re, "First: A framework for optimizing information quality in mobile crowdsensing systems," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 1, pp. 1–35, 2018.

[9] E. Casella, A. R. Khamesi, S. Silvestri, D. A. Baker, and S. K. Das, "Hvac power conservation through reverse auctions and machine learning," in *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2022, pp. 89–100.

[10] S. Ciavarella, J.-Y. Joo, and S. Silvestri, "Managing contingencies in smart grids via the internet of things," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 2134–2141, 2016.

[11] V. Dolce, C. Jackson, S. Silvestri, D. Baker, and A. De Paola, "Social-behavioral aware optimization of energy consumption in smart homes," in *2018 14th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2018, pp. 163–172.

[12] A. Timilsina and B. Paudyal, "Technical design of a grid-connected photovoltaic system and its challenges in nepalese power scenario," in *2017 7th International Conference on Power Systems (ICPS)*. IEEE, 2017, pp. 334–339.

[13] A. Khanal, A. Timilsina, B. Paudyal, and S. Ghimired, "Comparative analysis of cell balancing topologies in battery management systems," in *Proceedings of IOE Graduate Conference*, 2019.

[14] A. Timilsina, B. Nepali, B. Paudyal, and J. D. Kunwar, "Design and performance analysis of highly efficient class e resonant inverter," *Zerone Scholar*, vol. 1, no. 1, pp. 14–17, 2016.

[15] A. Timilsina, B. Paudyal, B. Nepali, J. D. Kunwar, A. K. Mishra, and S. K. Ghimire, "Analysis and optimization of magnetic resonant wireless power transfer system," in *Proceedings of IOE Graduate Conference*, vol. 5, 2017, p. 259.

[16] A. Yassine, M. S. Hossain, G. Muhammad, and M. Guizani, "Cloudlet-based intelligent auctioning agents for truthful autonomous electric vehicles energy crowdsourcing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5457–5466, 2020.

[17] Y. Zhao, K. Zheng, H. Yin, G. Liu, J. Fang, and X. Zhou, "Preference-aware task assignment in spatial crowdsourcing: from individuals to groups," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 7, pp. 3461–3477, 2020.

[18] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2016, pp. 344–353.

[19] A. Timilsina and S. Silvestri, "e-uber: A crowdsourcing platform for electric vehicle-based ride-and energy-sharing," *arXiv preprint arXiv:2304.04753*, 2023.

[20] A. A. Lazar and N. Semret, "The progressive second price auction mechanism for network resource sharing," in *8th International Symposium on Dynamic Games, Maastricht, The Netherlands*, 1998.

[21] J. Csirik, "Heuristics for the 0-1 min-knapsack problem," *Acta Cybernetica*, vol. 10, no. 1-2, pp. 15–20, 1991.

[22] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1466–1478, Oct 2012.

[23] R. M. Karp, "An algorithm to solve the m× n assignment problem in expected time o (mn log n)," *Networks*, vol. 10, no. 2, pp. 143–152, 1980.

[24] C. of New York Taxi and L. Commission, "New york city taxi and limousine commission (tlc) trip record data of the year 2013," 2019. [Online]. Available: https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[25] P. S. Inc., 2019. [Online]. Available: www.pecanstreet.org

[26] Y. Liu, X. Xu, J. Pan, J. Zhang, and G. Zhao, "A truthful auction mechanism for mobile crowd sensing with budget constraint," *IEEE Access*, vol. 7, pp. 43 933–43 947, 2019.