Timeliness-Aware Incentive Mechanism for Vehicular Crowdsourcing in Smart Cities

Xianhao Chen, Lan Zhang, *Student Member, IEEE,* Yawei Pang, *Student Member, IEEE,* Bin Lin, *Member, IEEE,* and Yuguang Fang, *Fellow, IEEE*

Abstract—Vehicular crowdsourcing is a promising paradigm that takes advantage of powerful onboard capabilities of vehicles to perform various tasks in smart cities. To fulfill this vision, a well-designed incentive mechanism is essential to stimulate the participation of vehicles. In this paper, we propose a timeliness-aware incentive mechanism for vehicular crowdsourcing by taking vehicle's uncertain travel time into account. In view of the stochastic nature of traffic conditions, we derive a tractable expression for the probability distribution of task delay based on a discrete-time traffic model. By leveraging reverse auction framework, we model the utility of a service requester as a function in terms of *uncertain* task delay and incurred payment. To maximize the requester's utility under a budget constraint, we cast the mechanism design as a non-monotone submodular maximization problem over a knapsack constraint. Based on this formulation, we develop a truthful budgeted utility maximization auction (TBUMA), which is truthful, budget feasible, profitable, individually rational and computationally efficient. Through extensive trace-based simulations, we demonstrate the effectiveness of our proposed incentive mechanism.

Index Terms—Vehicular crowdsourcing, incentive mechanism, reverse auction, crowdsensing, edge computing.

1 Introduction

Smart cities are the development trends for future cities, which are expected to leverage advanced information and communication technologies to improve city operation and quality of life [2]. To realize this vision, a compelling idea that has recently attracted much attention is to exploit the under-utilized onboard resources of vehicles to complement the existing infrastructures in supporting the ubiquitous smart-city services [3], [4], [5]. As vehicles are becoming connected and autonomous, they will be endowed with significantly powerful sensing, communication, computing, storage, and intelligence (SCCSI) capabilities in the near future [3], [6], [7]. Therefore, the abundant onboard resources of vehicles, if harvested and managed properly, can be utilized to provision services in smart cities without significantly increasing deployment cost. For example, future vehicles can be directly employed to sense urban areas [4], provide edge computing services to ubiquitous resource-constrained devices [5], or carry and forward delay-tolerant data to intended locations [3], [7], all of which could relieve the burden of our existing infrastructure in a cost-effective way.

Despite the diversity of provisioned services, the paradigms that employ vehicles to serve users' requests can be generalized as vehicular crowdsourcing. A typical vehicular crowdsourcing system consists of a service requester and many participating vehicles [8], where the requester publishes a set of location-dependent tasks for the participating vehicles to perform. However, when performing tasks, vehicles have to consume the resources of onboard units, e.g., battery, CPU, wireless devices and storage, and therefore, vehicle owners may not be interested in participation unless they receive some incentives in return. Consequently, a well-designed incentive mechanism is of paramount importance to the success of vehicular crowdsourcing.

To incentivize the participation of vehicles, auction is a natural choice due to its efficiency in satisfying the individual requirements. However, the design of auction-based incentive mechanism for vehicular crowdsourcing is highly complicated. First, a well-designed auction mechanism should prevent the strategic gaming of participants, i.e., guaranteeing *truthfulness*. As demonstrated by economic theory, an auction mechanism failing to achieve truthfulness is vulnerable to market manipulation and may produce very poor outcome [9]. Second, since the service requester usually faces budget constraint in the real world, it is non-trivial to consider *budget feasibility* in mechanism design.

In addition, the timeliness of information collection, analysis and sharing is critical for many vehicular crowdsourcing applications, such as video surveillance for public

- *Xianhao Chen and Yuguang Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA.*
  E-mail: xianhaochen@ufl.edu, fang@ece.ufl.edu.
- *Lan Zhang is with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI 49931, USA.*
  E-mail: lanzhang@mtu.edu.
- *Yawei Pang is with the Department of IoT, School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China.*
  E-mail: yaweipang@gmail.com.
- *Bin Lin is with School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China*
  E-mail: binlin@dlmu.edu.cn.

This work extends our paper [1] accepted by IEEE GLOBECOM 2019. This work was partially supported by US National Science Foundation under grants CNS-1717736 and IIS-1722791. The work of Y. Fang was also partially supported by Natural Science Foundation of China under 61672106. The work of B. Lin was partially supported by National Natural Science Foundation of China under 61971083 and 51939001 and Dalian Science and Technology Innovation Fund under 2019J11CY015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3052963, IEEE Transactions on Mobile Computing

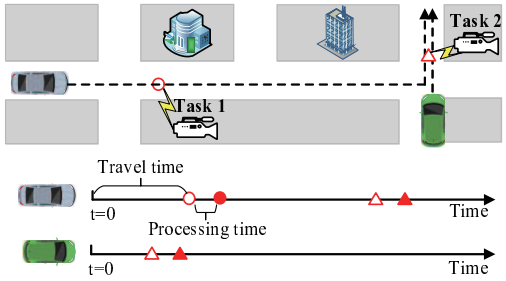


Fig. 1. An illustrative example of vehicular crowdsourcing with two tasks and two participating vehicles. Circle and triangle represent task 1 and task 2, respectively. In the time axes, the hallow icons denote the arrival times, the solid icons denote the completion times. When a task is assigned to a vehicle, the completion time is equal to the summation of the vehicle's travel time to the intended location and the processing time used to perform the task.

safety, parking availability detection, and popular content distribution, because the usefulness of tasks may decay over time. For example, when a potential crime is reported, a local police station needs to effectively locate the suspect. The wireless street cameras in places of interest can transmit the recorded videos to passing vehicles for video analytics [10], as illustrated in Fig. 1. Since the police station will take action according to the video analytics results, the tasks should be completed timely. As a result, an appropriate vehicle selection policy not only depends on vehicles' routes and recruitment costs as in [8], [11], but also depends on when the tasks are expected to complete.

Despite the importance of timeliness-aware incentive design, to our best knowledge, this problem has not been studied in the context of vehicular crowdsourcing by considering vehicle's travel time. Note that our considered scenario is fundamentally different from the case where nearby or parked vehicles are employed to provide computing services without leveraging their mobility [10], [12], [13]. Although recruiting nearby vehicle may yield the shortest delay and highest reliability, finding appropriate vehicles (e.g., asking for low prices) on site is not always possible. In general, vehicles demand distinct travel times to locations of interest. The difficulty of our timeliness-aware incentive design mainly comes from the fact that vehicle's travel time is inherently uncertain due to the dynamic nature of traffic conditions, which requires a novel auction scheme that captures the *uncertain service delay* and *the preference of service requester for shorter delay* while preserving the aforementioned essential economic properties. The key contributions of our work are summarized as follows.

- We model the interaction between a service requester and participating vehicles as a reverse auction. To factor timeliness into consideration, we characterize the requester's utility as a function in terms of both task delay and incurred payment. Different from existing auction schemes where each item has a constant valuation, an item (i.e., a task) in our auction model has a valuation function varying with delay. The proposed auction model is suitable to handling delay-aware incentive design, and more interestingly, applies to the case where delay is uncertain and follows general probability distribution.
- By considering the uncertainty of vehicle's travel time, we develop an efficient travel time distribution (TTD) estimation approach based on a discrete-time traffic model. Then, we obtain a tractable expression for the probability distribution of task delay in vehicular crowdsourcing context.
- To maximize requester's utility under a given budget, we cast the auction mechanism design as a non-monotone submodular maximization problem over a budget constraint. Maximizing the requester's utility is equivalent to achieving the optimal balance between uncertain task delay and incurred payment. Based on this formulation, we develop a truthful budgeted utility maximization auction (TBUMA), which guarantees truthfulness, individual rationality, profitability, budget feasibility, and computational efficiency.
- We conduct trace-based simulations to demonstrate the effectiveness of both the TTD estimation approach and the proposed incentive mechanism.

2 Related Work

2.1 Vehicle as a Service

The onboard resources of vehicles can be exploited to support a wide range of smart-city applications. On the one hand, vehicles can leverage their sensors to provide sensing services for the general public, which is called as vehicular crowdsensing [4], [8]. On the other hand, since autonomous vehicles have powerful onboard servers [14], by leveraging the underutilized computing capabilities when parked or slowly moving, vehicles can provide computing services to resource-limited end devices [5], [15].

The design of vehicular crowdsensing systems has attracted much attention [8], [11], [16], [17]. In [11], He et al. present a recruitment strategy for vehicular crowdsensing systems to maximize the sensing coverage. In [8], Gao et al. devise an incentive mechanism for vehicular crowdsensing sby taking uncertain trajectories of vehicles into consideration. In [16], Abdelhamid et al. develop a reputation-based vehicle selection strategy to maximize the sensing coverage. In [17], Xu et al. propose a vehicular crowdsensing system which incentivizes taxis to move to the trajectories that match the demand of service requester. Nevertheless, none of the aforementioned works have taken service delay into consideration. Another line of research leverages vehicles' computing capabilities to provide edge computing services [10], [12], [13], [18]. However, these works focus on the scenario where data sources can directly connect with nearby or parked vehicles for task offloading, thus not treating vehicle's uncertain travel time to intended location as a part of task delay.

2.2 Incentive Design for Mobile Crowdsourcing

Incentive design for mobile crowdsourcing has been extensively studied based on various game theoretical approaches [19]. In [20], Cheung et al. develop mobile crowdsourcing systems with deadline requirements by modeling the interactions between a requester and participants as a Stackelberg game. In [21], Zhan et al. study the incentive design

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3052963, IEEE Transactions on Mobile Computing

for time-sensitive cooperative data collection by formulating the problem as a two-user cooperative game.

In our system, each participating vehicle is allowed to select a set of location-dependent tasks to perform based on their own arrangements and preferences. Mathematically, the problem of participant selection falls in the category of combinatorial optimization. By considering incentive design, it is reasonable to model the vehicle selection as a reverse combinatorial auction. Along this line, Yang et al. propose an incentive mechanism for crowdsensing based on reverse combinatorial auction, which aims to maximize the utility of service requester [22]. In [23], Feng et al. propose a reverse auction scheme to stimulate smartphone users to join location-dependent crowdsensing with minimal social cost. In [24]. Restuccia et al. design a reverse auction scheme for mobile crowdsensing by considering budget limitation and uncertain mobility patterns of participants. However, their work does not rely on a vehicle-specific mobility model, and does not consider requester's preference for shorter service delay, which thus cannot apply to our scenario.

Some works in this domain consider budget feasibility [24], [25], [26]. Singla et al. develop a reverse auction mechanism to stimulate participants to share their private information [25]. Zhang et al. present a reverse auction framework for label collection in crowdsensing by jointly considering the budget constraint and the difficulty of labeling tasks [26]. Similar to [24], [25], [26], our algorithm also guarantees budget feasibility. However, unlike the aforementioned budget feasible mechanisms that aim to maximize total task valuation brought by recruited participants, our scheme is devised to maximize requester's utility, i.e., total task valuation minus incurred payment. Intuitively, the existing schemes always intend to run out their budgets, as their objective functions are unrelated to incurred payment. A bad consequence of this is that, they fail to guarantee *profitability*, which means that requester's utility may even be negative. In [27], Jiao et al. devise an auction scheme for public blockchain networks. Mathematically, their problem is a constrained non-monotone submodular maximization problem, which shares the same property with our problem. However, their constraint is for computing resources whereas our constraint is for payment. Thus, our budgeted mechanism is also fundamentally different from their scheme, because payments are potentially higher than bid prices and are not known in winner selection phase, which is the major challenge in guaranteeing budget feasibility.

3 System Architecture

3.1 Basic Model

As aforementioned, the considered vehicular crowdsourcing system is composed of a service requester and many participating vehicles. The interactions between a service requester and participating vehicles in this system is modeled as a single-round reverse auction, where the participating vehicles are the service sellers, the service requester is the service buyer. As a crowdsourcing campaign begins, the requester claims its *budget* B, and releases a set of location-dependent tasks, where each task contains *location*, *workload*, and *task valuation function*. Task valuation function will be further elaborated in Section 3.2.

TABLE 1
Frequently used notations

Notation	Description
I	the set of participating vehicles
J	the set of tasks
J_i	the set of tasks in vehicle i's bid
I^j	the set of vehicles bidding for task j
$\mathcal{P}_i$	vehicle i's path
l_i^n	the n-th link in vehicle i's path $\mathcal{P}_i$
$t_{i,j}^t$	vehicle i's travel time for performing task j, which is a random variable
$t_{i,j}^p$	vehicle i's processing time for task j, which is a constant
$t_{i,j}$	vehicle i's completion time for task j
$\mathcal{T}_i^p$	the set of processing times for J_i
$q_{i,j}^k$	the probability that vehicle i will complete task j in subinterval k
b_i	the bid price of vehicle i
c_i	the true cost of vehicle i
p_i	the payment to vehicle i
B	the budget of the service requester
v_j^k	the valuation of task j during the k-th subinterval
K	The subinterval number for task valuation
$\boldsymbol{\omega}$	the set of winners

Let $I = \{1, 2, ..., |I|\}$ be the set of participating vehicles, and $J = \{1, 2, ..., |J|\}$ be the set of tasks. Vehicle $i \in I$ chooses its path $\mathcal{P}_i$ and bids for a bundle of tasks $J_i \in J$ on the path by submitting a bid price b_i, i.e., the claimed lowest price at which vehicle i is willing to perform the set of tasks. Since each vehicle selects tasks on its chosen route, it is possible that $J_{i_1} \cap J_{i_2} \neq \emptyset$ for two vehicles i_1 and i_2. Vehicle i also has an associated true valuation c_i (which will be termed as true cost) for performing the subset of tasks J_i, which is privately known by itself. More concretely, true cost c_i is the minimum payment that vehicle i can accept to perform the subset of tasks J_i[1]. Assuming that all the vehicles are selfish but rational, they attempt to maximize their payoffs [22]. Thus, vehicle i may strategically manipulate bid price b_i to gain more payoff, so that b_i is not necessarily equal to true cost c_i. Besides, vehicle i should offer the set of processing times $\mathcal{T}_i^p = \{t_{i,j}^p | j \in J_i\}$, where $t_{i,j}^p$ represents vehicle i's processing time for task $j \in J_i$ after arrival at the required location. Notice that in our model, the time used by vehicle i to complete task j is the summation of processing time $t_{i,j}^p$ and travel time $t_{i,j}^t$. Processing time $t_{i,j}^p$ is estimated by vehicle i itself according to the task workload and its onboard resources, and should be submitted to the requester; travel time $t_{i,j}^t$ is directly estimated by the service requester based on vehicle i's planned path $\mathcal{P}_i$. The

1. In general, $c_i = d_i + \sum_{j \in J_i} c_{i,j}$, where $c_{i,j}$ is the compensation for vehicle i's onboard resource consumption for task j, and d_i is the compensation for the driver's effort (which is independent of task workloads, but is related to, say, the path $\mathcal{P}_i$ the driver chooses in order to perform the selected subset of tasks J_i). From the system viewpoint, we only care about the constant c_i, and therefore d_i and $c_{i,j}$ will not be used for the subsequent development.

requester is able to check whether $t^p_{i,j}$ is honestly reported by a vehicle after the vehicle arrives at the required location. Overall, the bid submitted by vehicle i can be characterized by a tuple $\beta_i = \{J_i, \mathcal{T}_i^p, b_i, \mathcal{P}_i\}$. For example, if vehicle i submits a bid $\beta_i = \{\{1,2\}, \{10,20\}, 0.3, \{l_i^1, l_i^2, l_i^3\}\}$, we can infer that vehicle i bids for task 1 and 2 with a bid price 0.3, its processing times for these two tasks are 10 and 20, respectively, and its chosen path contains three ordered links l_i^1, l_i^2, and l_i^3 that the vehicle will successively traverse. More details about road network and estimation of processing time $t^p_{i,j}$ and travel time $t^t_{i,j}$ will be provided in Section 3.3. The set of bids β_i from all the vehicles is denoted by $\mathcal{B}$. We will use vehicles and bidders interchangeably in the subsequent development. For readers' convenience, the frequently used notations in this paper are summarized in Table 1.

3.2 Timeliness-Aware Task Valuation

To ensure timeliness, one direction is to impose a hard deadline for each task. Based on this approach, however, a requester cannot distinguish two task delays as long as they are within the "maximum tolerable" deadline. To overcome this limitation, we model task valuation as a function in terms of delay. Each task has a valuation to the requester, which is equal to the maximum payment the requester is willing to pay in order to have it completed. Generally speaking, a service requester is willing to pay higher for a service with shorter delay, and vice versa, which implies that task valuation function is non-increasing with delay. As depicted in Fig. 2, we adopt a general discrete task valuation function, which partitions interval $[0, D]$ into K subintervals and remains constant within each subinterval:

$$v_j(t) = \begin{cases} v_j^k, & \text{if } T_{k-1} \le t \le T_k, \forall k \in [1, K], \\ 0, & \text{otherwise}, \end{cases} \quad (1)$$

where T_k denotes the upper bound of the k-th subinterval, with $T_K = D$ and $T_0 = 0$ by definition, and v_j^k is the valuation of task j during the k-th subinterval $[T_{k-1}, T_k]$, satisfying $v_j^1 \ge v_j^2 ... \ge v_j^K$. It is noted that K, v_j^k, and the length of each subinterval are all customized by the requester. In other words, the discrete function $v_j(t)$ can be in any shape to reflect the specific delay demand of a requester. As demonstrated in Section 5.4, as K increases, the computation complexity of our mechanism increases in $\mathcal{O}(K)$.

The discrete valuation function is convenient to quantify the contribution of a vehicle with uncertain travel time to a task. The bottom figure in Fig. 2 shows an illustrative example of $f_{t_{i,j}}(t)$, which is the probability density function (PDF) of completion time $t_{i,j}$, i.e., the time that vehicle i will complete task j. Clearly, given $f_{t_{i,j}}(t)$ and if vehicle i is the only vehicle assigned with task j, the valuation that vehicle i will contribute to task j is simply $\sum_{k=1}^K q_{i,j}^k v_j^k$, where $q_{i,j}^k$ is the integral of $f_{t_{i,j}}(t)$ over the range of $[T_{k-1}, T_k]$. Next, we will show how to derive $f_{t_{i,j}}(t)$ or $q_{i,j}^k$.

3.3 Estimation of Task Delay

As aforementioned, the completion time $t_{i,j}$ for vehicle i to complete task j is given by

$$t_{i,j} = t^p_{i,j} + t^t_{i,j}. \quad (2)$$

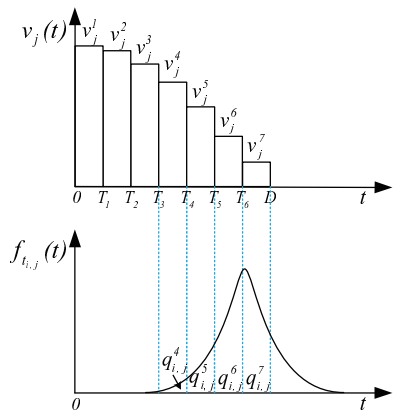


Fig. 2. An illustrative example of task valuation function and completion time distribution. The upper figure denotes the task valuation function of task j, and the bottom figure is the probability density function of $t_{i,j}$, i.e., $f_{t_{i,j}}(t)$, indicating when vehicle i can complete task j. $q_{i,j}^k$ is the probability that vehicle i will complete task j in time interval k.

In this paper, processing time $t^p_{i,j}$ is treated as a constant whereas travel time $t^t_{i,j}$ is a random variable due to the dynamics of traffic conditions. In what follows, we elaborate on how to calculate these two values, respectively.

3.3.1 Processing Time Estimation

The method of estimating processing time varies according to task type (e.g., sensing or computing task). Taking computing task as an example, $t^p_{i,j}$ can be expressed as

$$t^p_{i,j} = \frac{d_j}{r_j} + \frac{w_j}{f_{i,j}}, \quad (3)$$

where d_j is the input data size of task j, r_j is the estimated transmission rate from the corresponding data source to a vehicle, w_j is the computation workload of task j (in CPU cycles), and $f_{i,j}$ is vehicle i's CPU-cycle frequency allocated to task j [28]. Here, we assume that a data source transmits at a fixed data rate when vehicles enter its transmission range. Moreover, by assuming that the data size of computing result is negligible, we do not incorporate the time used by vehicle to return computing result into (3).

3.3.2 Travel Time Distribution Estimation

Due to the stochastic nature of real-time traffic conditions, travel time is essentially a random variable. The measurement of trip travel time distribution (TTD) between two points in a transportation network has attracted some attention [29]. Nevertheless, existing approaches to trip TTD estimation generally suffer from high computation complexity. In vehicular crowdsourcing systems, a requester has to estimate TTDs for all vehicle-task pairs during the auction stage, which poses stringent requirements on computational efficiency. In what follows, we present an efficient TTD estimation approach, which gives a tractable expression for travel time $t^t_{i,j}$. Our approach comprises two phases, i.e., offline link TTD esimation and online trip TTD prediction.

Phase 1: Offline Link TTD estimation: Link TTD estimation aims to estimate TTD for every link (road segment) in a road network, which is conducted based on historical traffic data before crowdsourcing campaign starts. Being consistent with [30], we adopt a discrete-time traffic model where

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3052963, IEEE Transactions on Mobile Computing

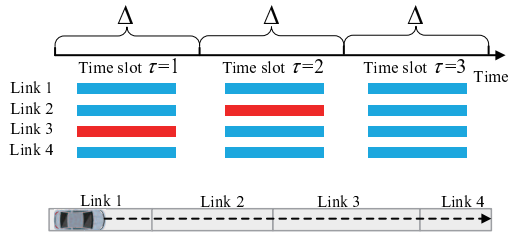


Fig. 3. Illustration of discrete-time traffic model. Due to the changing traffic state, it is assumed that the travel time for a link follows different normal distributions within different time slots. For illustrative purpose, in the figure, red and blue lines represent that the link is congested and undersaturated, respectively.

each link travel time follows a certain normal distribution conditional on time slot τ, as illustrated in Fig. 3. We assume that the crowdsourcing campaign starts at time 0, and use Δ, say 5 minutes, to denote the length of each time slot in the sense that time slot τ corresponds to time interval $[(\tau-1)\Delta, \tau\Delta]$. A smaller Δ leads to a finer discretization for traffic dynamics. Given historical data from probe vehicles or traffic sensors, the estimation of link TTDs for the upcoming time slots has been studied in [30], [31], [32]. Without loss of generality, we denote the TTD for link l in time slot τ by $\mathcal{N}(\mu_l^\tau, (\sigma_l^\tau)^2)$, and treat it as a known normal distribution in the subsequent development.

Phase 2: Online trip TTD estimation: Upon receiving bids from vehicles, the requester conducts trip TTD estimation for every vehicle-task pair. We assume that TTDs for different links are independent [32], and therefore trip travel time is equal to the aggregation of the link travel times on this trip. However, since the trip time is potentially greater than the length of time slot Δ, calculating a vehicle's trip TTD requires us to consider the time-varying link TTDs in the successive slots during a vehicle's trip. Recall that path $\mathcal{P}_i = [l_i^1, l_i^2, ..., l_i^{N_i}]$ is the set of ordered links traversed by vehicle i, where N_i is the number of links on path $\mathcal{P}_i$. To obtain the TTD for vehicle i traversing a given link on path $\mathcal{P}_i$, we should predict in which time slot it will enter this link, because the link TTD is dependent on time slot τ. Let $\rho_{i,n}^\tau$ denote the probability that vehicle i enters its n-th link on path $\mathcal{P}_i$ during time slot τ. It is known that any vehicle enters its first link during time slot 1, i.e., $\rho_{i,1}^1 = 1$, and $\rho_{i,1}^2 = \rho_{i,1}^3 = ... = 0$. Other $\rho_{i,n}^\tau$ for $n \geq 2$ are the parameters that we need to derive.

We denote x_i^n as the travel time for vehicle i to traverse its first n links in $\mathcal{P}_i$. Clearly, one can obtain $\rho_{i,n}^\tau$ as long as the distribution of x_i^{n-1} has been derived. Since vehicle i enters its first link during time slot $\tau = 1$, x_i^1 follows $\mathcal{N}\big(\alpha_i^1\mu_{l_i^1}^1, (\alpha_i^1\sigma_{l_i^1}^1)^2\big)$, where $\alpha_i^1 \in [0,1]$ is the ratio of the time vehicle i needs to traverse l_i^1 over the time needed to fully traverse this link. α_i^1 captures the fact that the first link may not be fully traversed by vehicle i, because vehicle i may locate at any position of its first link when the crowdsourcing starts. Supposing that a vehicle partially traverses link l_i^1 from location y_1 to location y_2 (which are distances to the end of the link), α_i^1 is equal to [30]

$$\alpha_i^1 = \int_{y_1}^{y_2} P_Y^{l_i^1}(y)dy, \tag{4}$$

where $P_Y^{l_i^1}(y)$ is the distribution for vehicles' locations within link l_i^1. In the case where vehicles are uniformly distributed within the link, α_i^1 is equal to the traversed length $|y_2 - y_1|$ over the entire link length.

For $n > 1$, x_i^n should account for the uncertainty of when vehicle i will enter its 2nd to nth links. To make our approach tractable, we use normal distribution $\mathcal{N}(\sum_{\tau=1}^{\tau_{\max}} \rho_{i,m}^\tau \mu_{l_i^m}^\tau, \sum_{\tau=1}^{\tau_{\max}} (\rho_{i,m}^\tau \sigma_{l_i^m}^\tau)^2)$ to approximate the travel time that vehicle i needs to traverse its m-th link l_i^m, where $\tau_{\max}$ is the index of time slot that is great enough for all vehicles traversing their paths. Then, by leveraging the additive property of normal distributions, we compute $x_i^n \sim \mathcal{N}(\mu_{x_i^n}, \sigma_{x_i^n}^2)$ by summing up all the link travel times:

$$\mu_{x_i^n} = \alpha_i^1 \mu_{l_i^1}^1 + \sum_{m=2}^{n} \sum_{\tau=1}^{\tau_{\max}} \rho_{i,m}^\tau \mu_{l_i^m}^\tau, \tag{5}$$

$$\sigma_{x_i^n}^2 = (\alpha_i^1 \sigma_{l_i^1}^1)^2 + \sum_{m=2}^{n} \sum_{\tau=1}^{\tau_{\max}} (\rho_{i,m}^\tau \sigma_{l_i^m}^\tau)^2. \tag{6}$$

Given that the vehicle starts its trip $\mathcal{P}_i$ at time 0, according to the definition, $\rho_{i,n}^\tau$ is determined by x_i^{n-1}:

$$\begin{aligned} \rho_{i,n}^\tau &= F_{x_i^{n-1}}(\tau\Delta) - F_{x_i^{n-1}}\big((\tau-1)\Delta\big) \\ &= \Phi\left(\frac{\tau\Delta - \mu_{x_i^{n-1}}}{\sigma_{x_i^{n-1}}}\right) - \Phi\left(\frac{(\tau-1)\Delta - \mu_{x_i^{n-1}}}{\sigma_{x_i^{n-1}}}\right), \end{aligned} \tag{7}$$

where $F_{x_i^{n-1}}(\cdot)$ is the cumulative distribution function (CDF) of x_i^{n-1}, and $\Phi(\cdot)$ represents the CDF of standard normal distribution. Notice that x_i^{n-1} can be used to compute $\rho_{i,n}^\tau$ based on (7), and then $\rho_{i,2}^\tau$ to $\rho_{i,n}^\tau$ can be further used to obtain x_i^n from (5) and (6). In this way, by starting from the second link of vehicle i, we can obtain $\rho_{i,n}^\tau$ for all the successive links on the trip $\mathcal{P}_i$. As a result, by employing the additive property again, the travel time for vehicle i to perform task j, i.e., $t_{i,j}^t$, follows normal distribution $\mathcal{N}(\mu_{t_{i,j}^t}, \sigma_{t_{i,j}^t}^2)$ with $\mu_{t_{i,j}^t} = \sum_{m=1}^{N_i} \sum_{\tau=1}^{\tau_{\max}} \gamma_{i,j}^m \rho_{i,m}^\tau \mu_{l_i^m}^\tau$ and $\sigma_{t_{i,j}^t}^2 = \sum_{m=1}^{N_i} \sum_{\tau=1}^{\tau_{\max}} (\gamma_{i,j}^m \rho_{i,m}^\tau \sigma_{l_i^m}^\tau)^2$. Analogous to α_i^1 in (4), $\gamma_{i,j}^m \in [0,1]$ accounts for whether and how long vehicle i needs to traverse link l_i^m in order to perform task j, as performing a specific task may not require vehicle i to fully traverse every link on its trip $\mathcal{P}_i$. Finally, by taking processing delay $t_{i,j}^p$ into account, the completion time $t_{i,j}$ follows $\mathcal{N}(\mu_{t_{i,j}}, \sigma_{t_{i,j}}^2)$ with

$$\mu_{t_{i,j}} = t_{i,j}^p + \sum_{m=1}^{N_i} \sum_{\tau=1}^{\tau_{\max}} \gamma_{i,j}^m \rho_{i,m}^\tau \mu_{l_i^m}^\tau, \tag{8}$$

$$\sigma_{t_{i,j}}^2 = \sum_{m=1}^{N_i} \sum_{\tau=1}^{\tau_{\max}} (\gamma_{i,j}^m \rho_{i,m}^\tau \sigma_{l_i^m}^\tau)^2. \tag{9}$$

Consequently, we can compute $q_{i,j}^k$ by

$$q_{i,j}^k = \begin{cases} \Phi\left(\dfrac{T_k - \mu_{t_{i,j}}}{\sigma_{t_{i,j}}}\right) - \Phi\left(\dfrac{T_{k-1} - \mu_{t_{i,j}}}{\sigma_{t_{i,j}}}\right), & \text{if } j \in J_i, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Recall that $q_{i,j}^k$ is the probability that vehicle i will complete task j in time interval k, as depicted in Fig. 2. $q_{i,j}^k$ is non-zero only if $j \in J_i$, i.e., task j is in vehicle i's bid. Since

the values of the $\Phi(\cdot)$ function has been tabulated, which are readily found, one can easily obtain $q_{i,j}^k$ from (10) for every vehicle-task pair in a large-scale crowdsourcing campaign.

Remark 1. *It is noted that our algorithm development will rely on generic $q_{i,j}^k$ values. Thus, one can also employ other TTD estimation methods with reasonable accuracy and complexity to compute $q_{i,j}^k$.*

3.4 Mathematical Formulation and Design Objectives

We next present the mathematical formulation and the design objectives of our mechanism. Given the bid information, we aim to determine a set of winners $\boldsymbol{\omega}$ from the bidding vehicles. To compute the expected valuation of task $j \in J$ brought by winner set $\boldsymbol{\omega}$, which is denoted by $V^j(\boldsymbol{\omega})$, one should consider the fact that there are potentially more than one winning vehicles bidding for task j. Intuitively, $V^j(\boldsymbol{\omega})$ is equal to the task valuation brought by the first vehicle completing it, which is given by

$$\begin{aligned} V^j(\boldsymbol{\omega}) &= \big(1 - Q_j^1(\boldsymbol{\omega})\big) v_j^1 + Q_j^1(\boldsymbol{\omega})\big(1 - Q_j^2(\boldsymbol{\omega})\big) v_j^2 \\ &\quad + ... + \Big(\prod_{k\in[2,K]} Q_j^{k-1}(\boldsymbol{\omega})\Big)\big(1 - Q_j^K(\boldsymbol{\omega})\big) v_j^K \\ &= \big(1 - Q_j^1(\boldsymbol{\omega})\big) v_j^1 \\ &\quad + \sum_{k=2}^{K} \Big(\prod_{m\in[2,k]} Q_j^{m-1}(\boldsymbol{\omega})\Big)\big(1 - Q_j^k(\boldsymbol{\omega})\big) v_j^k, \end{aligned} \tag{11}$$

with

$$Q_j^k(\boldsymbol{\omega}) = \prod_{i\in\boldsymbol{\omega}} (1 - q_{i,j}^k), \tag{12}$$

where v_j^k is given in (1), $q_{i,j}^k$ is given in (10), and $Q_j^k(\boldsymbol{\omega})$ represents the probability that there is no vehicle completing task j within the k-th time interval given the winner set $\boldsymbol{\omega}$. We can interpret (11) as follows. Recall that $v_j^1 \geq v_j^2 \geq ... \geq v_j^K$ in (1), and hence $V^j(\boldsymbol{\omega}) = v_j^k$ if and only if task j is not completed during the first $k-1$ time intervals, i.e., $\prod_{m\in[2,k]} Q_j^{m-1}(\boldsymbol{\omega}) = 1$, while being completed in the k-th interval, i.e., $Q_j^k(\boldsymbol{\omega}) = 0$.

To the service requester, the total valuation brought by winner set $\boldsymbol{\omega}$ is therefore expressed as

$$\mathcal{V}(\boldsymbol{\omega}) = \sum_{j\in J} V^j(\boldsymbol{\omega}). \tag{13}$$

The requester's utility is the total valuation brought by the winners minus the total payment:

$$\mathcal{R}(\boldsymbol{\omega}) = \mathcal{V}(\boldsymbol{\omega}) - \sum_{i\in\boldsymbol{\omega}} p_i. \tag{14}$$

The payoff of vehicle $i \in I$ is its received payment minus its true cost:

$$u_i = \begin{cases} p_i - c_i, & \text{if } i \in \boldsymbol{\omega}, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The social welfare is the summation of the requester's utility and the payoffs of all the participating vehicles [33]:

$$\mathcal{U}(\boldsymbol{\omega}) = \mathcal{V}(\boldsymbol{\omega}) - \sum_{i\in\boldsymbol{\omega}} c_i. \tag{16}$$

Our auction mechanism seeks a vehicle selection and pricing strategy to maximize the requester's utility. Maximizing the requester utility $\mathcal{R}(\boldsymbol{\omega})$ is equivalent to striking the optimal balance between task delay and incurred payment. Thus, we can formulate the following optimization problem

$$\begin{aligned} &\max_{\boldsymbol{\omega}\subseteq I, \boldsymbol{p}} \ \mathcal{V}(\boldsymbol{\omega}) - \sum_{i\in\boldsymbol{\omega}} p_i \\ &s.t. \quad \text{C1}: \sum_{i\in\boldsymbol{\omega}} p_i \leq B, \quad \text{C2}: p_i \geq b_i, \forall i \in \boldsymbol{\omega}, \end{aligned} \tag{17}$$

where $\boldsymbol{p}$ represents the collection of payment p_i, Constraint C1 ensures budget feasibility, and Constraint C2 guarantees individual rationality. We remark that the paths (or the set of bidding tasks) of vehicles, and the travel and processing times have been all converted to parameters $q_{i,j}^k$ (in the term $\mathcal{V}(\boldsymbol{\omega})$), which can be obtained from (10). Noticing that payment p_i is potentially unequal to bid price b_i, solving Problem (17) involves not only the selection of winners, but also the determination of payment p_i. Our mechanism needs to satisfy the following critical properties:

- *Individual Rationality:* Each bidder will have a non-negative payoff by offering its true cost (valuation) as the bid price.
- *Profitability:* For the requester, the total value brought by winners is not lower than the total payment given to them.
- *Computational Efficiency:* The algorithm will terminate in polynomial time.
- *Truthfulness:* $b_i = c_i$ is the dominant strategy for every bidder. In other words, no bidder can improve its payoff by claiming a bid price deviating from its true cost.
- *Budget Feasibility:* The total payment will not exceed the budget of the service requester.

The importance of truthfulness and budget feasibility has been discussed before. To stimulate drivers to participate, their payoffs must be non-negative. Without being profitable, the service requester would not be interested in outsourcing its tasks to participants. Computational efficiency guarantees that the mechanism is feasible in large-scale applications.

4 Budgeted Utility Maximization Auction (BUMA)

In this section, we prove that Problem (17) with fixed p_i is a constrained non-monotone submodular maximization problem. Then, we develop an approximation algorithm, called budgeted utility maximization auction (BUMA), to solve Problem (17).

It is clear that the simplest payment strategy $p_i = b_i$ can maximize the requester's utility while guaranteeing individual rationality. Let us focus on this payment strategy in this section. By setting $p_i = b_i$, Problem (17) is reformulated as

$$\max_{\boldsymbol{\omega}\subseteq I} \ \mathcal{V}(\boldsymbol{\omega}) - \sum_{i\in\boldsymbol{\omega}} b_i \quad s.t. \quad \text{C1}: \sum_{i\in\boldsymbol{\omega}} b_i \leq B, \tag{18}$$

To solve Problem (18), we first prove that its objective function is submodular.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3052963, IEEE Transactions on Mobile Computing

Definition 1. *(submodular). Given a set $\boldsymbol{V}$, a function $f: 2^{\boldsymbol{V}} \to \mathbb{R}$ is submodular if for all $\boldsymbol{A} \subseteq \boldsymbol{C} \subseteq \boldsymbol{V}$ and $i \in \boldsymbol{V} \setminus \boldsymbol{C}$, we have*

$$f(\boldsymbol{A} \cup \{i\}) - f(\boldsymbol{A}) \geq f(\boldsymbol{C} \cup \{i\}) - f(\boldsymbol{C}),$$

where $\mathbb{R}$ is the set of real numbers.

Theorem 1. *The objective function of problem (17) is submodular.*

Proof. By Definition 1, proving the submodularity of $\mathcal{R}(\cdot)$ is equivalent to showing that $\mathcal{R}(\boldsymbol{A} \cup \{i\}) - \mathcal{R}(\boldsymbol{A}) \geq \mathcal{R}(\boldsymbol{C} \cup \{i\}) - \mathcal{R}(\boldsymbol{C})$, for all $\boldsymbol{A} \subseteq \boldsymbol{C} \subseteq I$ and $i \in I \setminus \boldsymbol{C}$. By removing the payment terms from both sides, we only need to prove that $\mathcal{V}(\boldsymbol{A} \cup \{i\}) - \mathcal{V}(\boldsymbol{A}) \geq \mathcal{V}(\boldsymbol{C} \cup \{i\}) - \mathcal{V}(\boldsymbol{C})$. Since $\mathcal{V}(\boldsymbol{A}) = \sum_{j \in J} V^j(\boldsymbol{A})$, it is sufficient to show that $V^j(\boldsymbol{A} \cup \{i\}) - V^j(\boldsymbol{A}) \geq V^j(\boldsymbol{C} \cup \{i\}) - V^j(\boldsymbol{C})$ for all $j \in J$.

By the definition of $V^j(\boldsymbol{\omega})$ in (11), we have

$$\begin{aligned}
&V^j(\boldsymbol{A} \cup \{i\}) - V^j(\boldsymbol{A}) = \big(1 - Q_j^1(\boldsymbol{A} \cup \{i\})\big) v_j^1 \\
&+ \sum_{k=2}^{K} \Big(\prod_{m \in [2,k]} Q_j^{m-1}(\boldsymbol{A} \cup \{i\}) \Big) \big(1 - Q_j^k(\boldsymbol{A} \cup \{i\})\big) v_j^k \\
&- \big(1 - Q_j^1(\boldsymbol{A})\big) v_j^1 - \sum_{k=2}^{K} \Big(\prod_{m \in [2,k]} Q_j^{m-1}(\boldsymbol{A}) \Big) \big(1 - Q_j^k(\boldsymbol{A})\big) v_j^k \\
&= \big(1 - Q_j^1(\boldsymbol{A})(1 - q_{i,j}^1)\big) v_j^1 \\
&+ \sum_{k=2}^{K} \Big(\prod_{m \in [2,k]} Q_j^{m-1}(\boldsymbol{A})(1 - q_{i,j}^{m-1}) \Big) \big(1 - Q_j^k(\boldsymbol{A})(1 - q_{i,j}^k)\big) v_j^k \\
&- \big(1 - Q_j^1(\boldsymbol{A})\big) v_j^1 - \sum_{k=2}^{K} \Big(\prod_{m \in [2,k]} Q_j^{m-1}(\boldsymbol{A}) \Big) \big(1 - Q_j^k(\boldsymbol{A})\big) v_j^k.
\end{aligned}$$

After some manipulations, we obtain

$$\begin{aligned}
&V^j(\boldsymbol{A} \cup \{i\}) - V^j(\boldsymbol{A}) \\
&= \Big(\prod_{k=1}^{K} Q_j^k(\boldsymbol{A}) \Big) \Big(1 - \prod_{k=1}^{K} (1 - q_{i,j}^k)\Big) v_j^K \\
&+ \sum_{k=1}^{K-1} \Big(\prod_{m=1}^{k} Q_j^m(\boldsymbol{A}) \Big) \Big(1 - \prod_{m=1}^{k} (1 - q_{i,j}^m)\Big) \big(v_j^k - v_j^{k+1}\big) \\
&\geq \Big(\prod_{k=1}^{K} Q_j^k(\boldsymbol{C}) \Big) \Big(1 - \prod_{k=1}^{K} (1 - q_{i,j}^k)\Big) v_j^K \\
&+ \sum_{k=1}^{K-1} \Big(\prod_{m=1}^{k} Q_j^m(\boldsymbol{C}) \Big) \Big(1 - \prod_{m=1}^{k} (1 - q_{i,j}^m)\Big) \big(v_j^k - v_j^{k+1}\big) \\
&= V^j(\boldsymbol{C} \cup \{i\}) - V^j(\boldsymbol{C}), \qquad (19)
\end{aligned}$$

where the above inequality follows from $Q_j^k(\boldsymbol{A}) \geq Q_j^k(\boldsymbol{C})$. The proof is completed. □

Similarly, we can prove that $\mathcal{U}(\boldsymbol{\omega})$ and $\mathcal{V}(\boldsymbol{\omega})$ are submodular as well. Moreover, since the requester's utility does not monotonically increases with the winner set $\boldsymbol{\omega}$ due to the incurred payment $\sum_{i \in \boldsymbol{\omega}} b_i$, Problem (18) is a non-monotone submodular maximization problem over a knapsack constraint, which is NP-hard [34]. There are several algorithms in the literature that could solve this problem, and we use the $\frac{1}{7+\epsilon}$-approximation algorithm developed in [35] due to its low computational complexity. However, the existing algorithms, including the one in [35], require the objective function to be non-negative, whereas the objective function in Problem (18) can be negative. To address this challenge, let $f(\boldsymbol{\omega}) = \mathcal{R}(\boldsymbol{\omega}) + \sum_{i \in I} b_i$. Since $\sum_{i \in I} b_i$ is a constant, $f(\boldsymbol{\omega})$ remains submodular. By using $f(\boldsymbol{\omega})$ as the objective function, we develop our algorithm based on the algorithm of [35], which is called BUMA. As summarized in Algorithm 1, BUMA has a multi-pass structure to construct three candidate solutions. We briefly introduce subroutine $Greedy_3$ and $LocalSearch$ in Algorithm 1 as follows.

Algorithm 1 BUMA Mechanism

Input: Set of tasks J, task valuation functions $v_j(t)$'s, set of bidders I, and $q_{i,j}^k$'s.
Output: Set of winners $\boldsymbol{\omega}$

```
Initialize ω = ∅, X_1 = I;
for k = 1 to 2 do
    S_k = Greedy_3(X_k);
    if k = 1 then
        S'_k = LocalSearch(S_k, ε);
        X_{k+1} = X_k \ S_k;
    end if
end for
ω = arg max_{S ∈ {S_1, S'_1, S_2}} f(S);
```

$Greedy_3$ first enumerates the subsets of I of cardinality $1, 2, 3$. Then, for each subset of cardinality 3, $Greedy_3$ extends it by greedily selecting the element i with the maximum $f_i(\boldsymbol{\omega}) = \frac{f(\boldsymbol{\omega} \cup \{i\}) - f(\boldsymbol{\omega})}{b_i}$ while satisfying the budget constraint, where $\boldsymbol{\omega}$ is the set of already selected elements. The greedy procedure terminates as the maximum $f_i(\boldsymbol{\omega}) \leq 0$ or all bidders are selected. The solution is the best from all sets of cardinality at most three that meets the budget constraint, as well as the greedy extensions of the sets of cardinality three. The detail of $Greedy_3$ can be found in Appendix B of [35].

$LocalSearch$ is a local search technique for unconstrained non-monotone submodular maximization problems. After removing Constraint C1, $LocalSearch$ searches for a better solution by iteratively adding or deleting one element such that $f(\boldsymbol{\omega} \cup \{i\}) > (1 + \frac{\epsilon}{|I|^2}) f(\boldsymbol{\omega})$ or $f(\boldsymbol{\omega} \setminus \{i\}) > (1 + \frac{\epsilon}{|I|^2}) f(\boldsymbol{\omega})$ whenever possible, where $\epsilon > 0$. The detail of $LocalSearch$ can be found in [36].

It is easy to prove that BUMA achieves all the desirable properties mentioned in Section 3.4 except truthfulness. To demonstrate that BUMA is untruthful, let us consider a toy example that there are two bidders (bidder 1 and 2) bidding for three identical tasks (task 1, 2 and 3). We set budget B to 1. As for the task valuation functions, we set $D = 200$, where $[0, D]$ is partitioned into 2 equal subintervals, satisfying $v_j^1 = 1$, $v_j^2 = 0.5$, $\forall j \in \{1, 2, 3\}$. Assume that bidder 1 bids for task 1 and task 2 with bid price 0.7, and bidder 2 bids for task 2 and task 3 with bid price 0.8. Moreover, suppose that we obtain $q_{1,1}^1 = 0.4$, $q_{1,1}^2 = 0.6$, $q_{1,2}^1 = 0.2$, $q_{1,2}^2 = 0.7$, $q_{2,2}^1 = 0.5$, $q_{2,2}^2 = 0.3$, $q_{2,3}^1 = 0$, and $q_{2,3}^2 = 0.6$. In this case, bidder 1 is selected as the only winner and its received payment is $p_1 = b_1 = 0.7$. However, when bidder 1 raises its bid price to $0.7 + \gamma$, where $\gamma \leq 0.285$, it is still selected as the only winner, whereas payment p_1 increases to $0.7 + \gamma$. Therefore, a bidder can increase its payoff by

offering a bid price unequal to its real cost.

Although BUMA approximately maximizes the requester's utility under a budget constraint, the failure of ensuring truthfulness makes it less attractive. Thus, BUMA will only be adopted as the benchmark to examine the approximation performance of the truthful auction mechanism presented in the next section.

5 Truthful Budgeted Utility Maximization Auction (TBUMA)

In the following, we develop an approximation algorithm to solve Problem (17) while guaranteeing the truthfulness, which is named as TBUMA mechanism. TBUMA consists of two steps, i.e., the winner selection procedure and the payment determination procedure.

5.1 Winner Selection Procedure

We first define marginal valuation, i.e., the incremental valuation that a new vehicle can contribute. Given existing winner set $\boldsymbol{\omega}$, the marginal value brought by vehicle i is

$$\mathcal{V}_i(\boldsymbol{\omega}) = \mathcal{V}(\boldsymbol{\omega} \cup \{i\}) - \mathcal{V}(\boldsymbol{\omega}). \tag{20}$$

Similarly, given existing winner set $\boldsymbol{\omega}$, we define the marginal social welfare brought by i as $\mathcal{U}_i(\boldsymbol{\omega}) = \mathcal{V}_i(\boldsymbol{\omega}) - c_i$. However, true cost c_i is unavailable to the service requester. Fortunately, since our mechanism is truthful (which will be proven in Section 5.4), i.e., bidding $b_i = c_i$ is the dominant strategy for all the bidders, we can use $\mathcal{V}_i(\boldsymbol{\omega}) - b_i$ instead to denote the marginal social welfare, i.e.,

$$\mathcal{U}_i(\boldsymbol{\omega}) = \mathcal{V}_i(\boldsymbol{\omega}) - b_i. \tag{21}$$

As presented in Algorithm 2, the winner selection procedure first compares budget B with $\mathcal{V}(I)$. If $B < \mathcal{V}(I)$, the algorithm searches for bidder i with the maximum $\frac{\mathcal{U}_i(\boldsymbol{\omega})}{b_i}$ in line 3 (or equivalently the maximum $\frac{\mathcal{V}_i(\boldsymbol{\omega})}{b_i}$, as $\frac{\mathcal{U}_i(\boldsymbol{\omega})}{b_i} = \frac{\mathcal{V}_i(\boldsymbol{\omega})}{b_i} - 1$), and check whether it satisfies the *budget feasible criterion* in line 6. If yes, this bidder (termed as the candidate bidder) is added to the winner set, and the procedure goes to the next iteration. Otherwise, the algorithm discards this bidder and searches for other candidate bidders (line 3 to line 11). Line 6 is adapted from the criterion designed for monotone submodular functions [37]. As proven in Section 5.4, while payment p_i is not determined at this stage, the criterion in line 6 strictly guarantees the budget feasibility. If $B \geq \mathcal{V}(I)$, the algorithm greedily selects the winners without checking the budget feasible criterion (line 13 to line 19), as it is shown in Section 5.4 that the budget feasibility is naturally satisfied in this condition. For both cases, the selection procedure terminates when no bidder is associated with a positive marginal social welfare (line 4 or line 14). For ease of presentation, we denote the n-th winner (in the order of selection) by bidder w_n, the winner set including the first n winners by $\boldsymbol{\omega}_n = \{w_1, w_2, ..., w_n\}$, and the final winner set by $\boldsymbol{\omega}$ in Algorithm 2 and the subsequent development.

Remark 2. *Since p_i has not been determined at the current stage, TBUMA greedily selects the winner with the maximum $\frac{\mathcal{U}_i(\boldsymbol{\omega}_{n-1})}{b_i}$ (or equivalently $\frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{b_i}$). Therefore, TBUMA approximately maximizes both requester's utility and social welfare.*

Algorithm 2 Winner Selection Procedure

Input: Set of tasks J, task valuation functions $v_j(t)$'s, set of bidders I, and $q_{i,j}^k$'s.
Output: Set of winners $\boldsymbol{\omega}$.

1: Initialize $\boldsymbol{\omega}_0 = \emptyset$, $n = 1$;
2: **if** $B < \mathcal{V}(I)$ **then**
3: $i' = \arg\max_{i \in I} \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{b_i}$;
4: **while** $\mathcal{V}_{i'}(\boldsymbol{\omega}_{n-1}) > b_{i'}$ and $I \neq \emptyset$ **do**
5: $I = I \setminus \{i'\}$;
6: **if** $b_{i'} \leq \frac{B}{\min\{2, \frac{\mathcal{V}(I)}{B}\}} \cdot \frac{\mathcal{V}_{i'}(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega}_{n-1} \cup \{i'\})}$ **then**
7: $\boldsymbol{\omega}_n = \boldsymbol{\omega}_{n-1} \cup \{i'\}$;
8: $n = n + 1$;
9: **end if**
10: $i' = \arg\max_{i \in I} \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{b_i}$;
11: **end while**
12: **else**
13: $i' = \arg\max_{i \in I} \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{b_i}$;
14: **while** $\mathcal{V}_{i'}(\boldsymbol{\omega}_{n-1}) > b_{i'}$ and $I \neq \emptyset$ **do**
15: $I = I \setminus \{i'\}$;
16: $\boldsymbol{\omega}_n = \boldsymbol{\omega}_{n-1} \cup \{i'\}$;
17: $n = n + 1$;
18: $i' = \arg\max_{i \in I} \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{b_i}$;
19: **end while**
20: **end if**

5.2 Payment Determination Procedure

During payment determination procedure, each winner is paid with *critical value*, i.e., the highest bid price that can make it win. As illustrated in Algorithm 3, this procedure determines the payment to winner $i \in \boldsymbol{\omega}$ by executing the winner selection procedure in the set $I_i = I \setminus \{i\}$. For the winner selection for I_i, we denote the m-th winner by w_m^i, the winner set including the first m winners by $\boldsymbol{\omega}_m^i = \{w_1^i, w_1^i, ..., w_m^i\}$, the final winner set by $\boldsymbol{\omega}^i$, and the total number of winners by M.

We first present the payment determination scheme when $B < \mathcal{V}(I)$ (line 4 to line 17). According to Algorithm 2, the bid price that bidder i can make to win against winner w_m^i should satisfy $\frac{\mathcal{V}_i(\boldsymbol{\omega}_{m-1}^i)}{b_i} \geq \frac{\mathcal{V}_{w_m^i}(\boldsymbol{\omega}_{m-1}^i)}{b_{w_m^i}}$, i.e.,

$$b_i \leq \rho_m^i = \frac{\mathcal{V}_i(\boldsymbol{\omega}_{m-1}^i) b_{w_m^i}}{\mathcal{V}_{w_m^i}(\boldsymbol{\omega}_{m-1}^i)}. \tag{22}$$

Besides, bidder i also needs to meet the budget feasible criterion, i.e.,

$$b_i \leq \nu_m^i = \frac{B}{\min\{2, \frac{\mathcal{V}(I)}{B}\}} \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{m-1}^i)}{\mathcal{V}(\boldsymbol{\omega}_{m-1}^i \cup \{i\})}, \tag{23}$$

Combining (22) and (23), it follows that the highest bid price that i can declare to win against the m-th winner is $\min\{\rho_m^i, \nu_m^i\}$ (line 10). In addition, after the M winners are selected, there are no bidders that can win against i. Thus, the bid price that makes bidder i become the $(M+1)$-th winner should meet

$$b_i \leq \max\left\{\mathcal{V}_i(\boldsymbol{\omega}_M^i), \nu_{M+1}^i\right\}, \tag{24}$$

as bidder i still needs to satisfy $\mathcal{V}_i(\boldsymbol{\omega}_M^i) \geq b_i$ and the budget feasible criterion. Finally, the critical value is set to

Algorithm 3 Payment Determination Procedure

Input: Set of tasks J, task valuation functions $v_j(t)$'s, set of bidders I, set of winners $\boldsymbol{\omega}$, and $q_{i,j}^k$'s.
Output: Payment p_i to winner $i \in \boldsymbol{\omega}$.

1: **for all** $i \in \boldsymbol{\omega}$ **do**
2: Initialize $\boldsymbol{\omega}_0^i = \emptyset$, $p_i = 0$, $I_i = I \setminus \{i\}$, $m = 1$;
3: **if** $B < \mathcal{V}(I)$ **then**
4: i'= $\arg\max_{x \in I_i} \frac{\mathcal{V}_x(\boldsymbol{\omega}_{m-1}^i)}{b_x}$;
5: **while** $\mathcal{V}_{i'}(\boldsymbol{\omega}_{m-1}^i) > b_{i'}$ and $I_i \neq \emptyset$ **do**
6: $I_i = I_i \setminus \{i'\}$;
7: **if** $b_{i'} \leq \frac{B}{\min\{2, \frac{\mathcal{V}(I)}{B}\}} \cdot \frac{\mathcal{V}_{i'}(\boldsymbol{\omega}_{m-1}^i)}{\mathcal{V}(\boldsymbol{\omega}_{m-1}^i \cup \{i'\})}$ **then**
8: $\rho_m^i = \frac{\mathcal{V}_i(\boldsymbol{\omega}_{m-1}^i) b_{i'}}{\mathcal{V}_{i'}(\boldsymbol{\omega}_{m-1}^i)}$;
9: $\nu_m^i = \frac{B}{\min\{2, \frac{\mathcal{V}(I)}{B}\}} \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{m-1}^i)}{\mathcal{V}(\boldsymbol{\omega}_{m-1}^i \cup \{i\})}$;
10: $p_i = \max\{p_i, \min\{\rho_m^i, \nu_m^i\}\}$;
11: $\boldsymbol{\omega}_m^i = \boldsymbol{\omega}_{m-1}^i \cup \{i'\}$;
12: $m = m + 1$;
13: **end if**
14: i'= $\arg\max_{x \in I_i} \frac{\mathcal{V}_x(\boldsymbol{\omega}_{m-1}^i)}{b_x}$;
15: **end while**
16: $\nu_m^i = \frac{B}{\min\{2, \frac{\mathcal{V}(I)}{B}\}} \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{m-1}^i)}{\mathcal{V}(\boldsymbol{\omega}_{m-1}^i \cup \{i\})}$;
17: $p_i = \max\{p_i, \min\{\mathcal{V}_i(\boldsymbol{\omega}_{m-1}^i), \nu_m^i\}\}$;
18: **else**
19: i'= $\arg\max_{x \in I_i} \frac{\mathcal{V}_x(\boldsymbol{\omega}_{m-1}^i)}{b_x}$;
20: **while** $\mathcal{V}_{i'}(\boldsymbol{\omega}_{m-1}^i) > b_{i'}$ and $I_i \neq \emptyset$ **do**
21: $I_i = I_i \setminus \{i'\}$;
22: $\rho_m^i = \frac{\mathcal{V}_i(\boldsymbol{\omega}_{m-1}^i) b_{i'}}{\mathcal{V}_{i'}(\boldsymbol{\omega}_{m-1}^i)}$;
23: $p_i = \max\{p_i, \rho_m^i\}$;
24: $\boldsymbol{\omega}_m^i = \boldsymbol{\omega}_{m-1}^i \cup \{i'\}$;
25: $m = m + 1$;
26: i'= $\arg\max_{x \in I_i} \frac{\mathcal{V}_x(\boldsymbol{\omega}_{m-1}^i)}{b_x}$;
27: **end while**
28: $p_i = \max\{p_i, \mathcal{V}_i(\boldsymbol{\omega}_{m-1}^i)\}$;
29: **end if**
30: **end for**

the maximum of these $M+1$ prices (line 17), i.e.,

$$p_i = \max\Big\{ \max_{1 \leq m \leq M} \{ \min\{\rho_m^i, \nu_m^i\}\}, \min\{\mathcal{V}_i(\boldsymbol{\omega}_M^i), \nu_{M+1}^i\}\Big\}. \tag{25}$$

When $B \geq \mathcal{V}(I)$ (line 19 to line 28), the payment determination strategy runs the winner selection procedure without checking the budget feasible criterion. Similar to (25), we have

$$p_i = \max\Big\{ \max_{1 \leq m \leq M} \rho_m^i, \mathcal{V}_i(\boldsymbol{\omega}_M^i)\Big\}, \tag{26}$$

which corresponds to line 28.

5.3 An illustrative Example

We use a walk-through example with four tasks and four vehicles in Fig. 4 to illustrate how TBUMA works. For task $j \in [1,4]$, we set the deadline $D = 200$, and partition the interval $[0, D]$ into $K = 5$ subintervals, with task valuation $v_j^1 = 1$, $v_j^2 = 0.8$, $v_j^3 = 0.6$, $v_j^4 = 0.4$, and $v_j^5 = 0.2$ in (1). Budget B is set to 3. The arrows plotted in Fig. 4 represent which tasks are bid by each vehicle, and the bid prices from vehicle 1 to vehicle 4 are 0.8, 0.8, 0.5, and 0.5, respectively. Moreover, we assume that the mean of completion time $\mu_{t_{1,1}} = 40$, $\mu_{t_{1,2}} = 120$, $\mu_{t_{1,3}} = 130$, $\mu_{t_{2,2}} = 60$, $\mu_{t_{2,4}} = 80$, $\mu_{t_{3,3}} = 20$, and $\mu_{t_{4,4}} = 80$, and the standard deviations $\sigma_{t_{i,j}} = 0.05\mu_{t_{i,j}}, \forall i \in [1,4], j \in [1,4]$ for illustrative purpose. Through (10), we can obtain $q_{1,1}^1 = 0.5$, $q_{1,1}^2 = 0.5$, $q_{1,2}^3 = 0.5$, $q_{1,2}^4 = 0.5$, $q_{1,3}^3 = 0.062$, $q_{1,3}^4 = 0.938$, $q_{2,2}^2 = 1$, $q_{2,4}^4 = 0.0131$, $q_{2,4}^5 = 0.9737$, $q_{3,3}^1 = 1$, $q_{4,4}^2 = 0.5$, and $q_{4,4}^3 = 0.5$, while the remaining $q_{i,j}^k$'s not listed here are approximately equal to zero. Then, we find that $\mathcal{V}(\{1,2,3,4\}) = \sum_{j=1}^4 V^j(\{1,2,3,4\}) = 3.0994 > B = 3$. Thus, we use line 3-11 in Algorithm 2 to select winners, and line 4-17 in Algorithm 3 to determine payments.

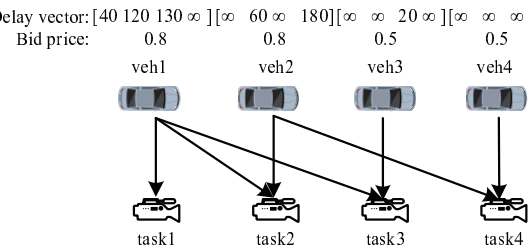


Fig. 4. An illustrative example of TBUMA with four tasks and four vehicles. The delay vector [task1 task2 task3 task4] denotes the mean of time that vehicle will use to complete each task, where ∞ corresponds to a task that the vehicle does not bid for.

Winner selection procedure (**Algorithm 2**):

$\boldsymbol{\omega}_0 = \emptyset$: $\frac{\mathcal{V}_1(\boldsymbol{\omega}_0)}{b_1} = \frac{\sum_{j \in \{1,2,3\}} V^j(\{1\} \cup \emptyset) - V^j(\emptyset)}{0.8} = 1.8614$, $\frac{\mathcal{V}_2(\boldsymbol{\omega}_0)}{b_2} = \frac{\sum_{j \in \{2,4\}} V^j(\{2\} \cup \emptyset) - V^j(\emptyset)}{0.8} = 1.2468$, $\frac{\mathcal{V}_3(\boldsymbol{\omega}_0)}{b_3} = \frac{V_3(\{3\} \cup \emptyset) - V_3(\emptyset)}{0.5} = 2$, $\frac{\mathcal{V}_4(\boldsymbol{\omega}_0)}{b_4} = \frac{V_4(\{4\} \cup \emptyset) - V_4(\emptyset)}{0.5} = 1.1$. The maximum value $\frac{\mathcal{V}_3(\boldsymbol{\omega}_0)}{b_3} = 2$ (line 3). Since $\mathcal{V}_3(\boldsymbol{\omega}_0) > b_3$ (line 4) and $b_3 = 0.5 \leq \frac{3}{\min\{2, \frac{3.0994}{3}\}} \cdot \frac{1}{1} = 2.9037$ (line 6), vehicle 3 wins.

$\boldsymbol{\omega}_1 = \{3\}$: $\frac{\mathcal{V}_1(\boldsymbol{\omega}_1)}{b_1} = \frac{\sum_{j \in \{1,2,3\}} V^j(\{1\} \cup \{3\}) - V^j(\{3\})}{0.8} = 1.375$. Likewise, $\frac{\mathcal{V}_2(\boldsymbol{\omega}_1)}{b_2} = 1.2468$, $\frac{\mathcal{V}_4(\boldsymbol{\omega}_1)}{b_4} = 1.1$. The maximum value $\frac{\mathcal{V}_1(\boldsymbol{\omega}_1)}{b_1} = 1.375$. Since $\mathcal{V}_1(\boldsymbol{\omega}_1) > b_1$ and $b_1 = 0.8 \leq \frac{3}{\min\{2, \frac{3.0994}{3}\}} \cdot \frac{1.1}{2.1} = 1.5211$, vehicle 1 wins.

$\boldsymbol{\omega}_2 = \{3,1\}$: $\frac{\mathcal{V}_2(\boldsymbol{\omega}_2)}{b_2} = 0.7468$, $\frac{\mathcal{V}_4(\boldsymbol{\omega}_2)}{b_4} = 1.1$. The maximum value $\frac{\mathcal{V}_4(\boldsymbol{\omega}_2)}{b_4} = 1.1$. Since $\mathcal{V}_4(\boldsymbol{\omega}_2) > b_4$ and $b_4 = 0.5 \leq \frac{3}{\min\{2, \frac{3.0994}{3}\}} \cdot \frac{0.55}{2.65} = 0.6027$, vehicle 4 wins.

$\boldsymbol{\omega}_3 = \{3,1,4\}$: $\frac{\mathcal{V}_2(\boldsymbol{\omega}_3)}{b_2} = 0.5617$. Since $\mathcal{V}_2(\boldsymbol{\omega}_3) < b_2$, the selection procedure terminates.

Payment determination procedure (**Algorithm 3**):

Since payment scheme runs for each winner independently, we only show how to determine the payment for winner 1. By excluding vehicle 1, we run the winner selection for vehicle set $\{2,3,4\}$ by using the winner selection procedure described above. Due to the space limitation, we directly give the winner selection results: $\boldsymbol{w}_0^1 = \emptyset$, $\boldsymbol{w}_1^1 = \{3\}$, and $\boldsymbol{w}_2^1 = \{3,2\}$ (line 11). Thus, we have $\min\{\rho_1^1, \nu_1^1\} = \min\{\frac{1.4891 \times 0.5}{1}, \frac{3}{\min\{2, \frac{3.0994}{3}\}} \frac{1.8614}{1.8614}\} = 0.7446$ (line 10), $\min\{\rho_2^1, \nu_2^1\} = \min\{\frac{1.1 \times 0.8}{0.9974}, \frac{3}{\min\{2, \frac{3.0994}{3}\}} \frac{1.1}{1+1.1}\} = 0.8823$ (line 10), and $\min\{\mathcal{V}_1(\boldsymbol{w}_2^1), \nu_3^1\} = \min\{0.7, \frac{3}{\min\{2, \frac{3.0994}{3}\}} \frac{0.7}{2.6974}\} = 0.7$ (line 17). The payment

for vehicle 1 is therefore the maximum of these values, i.e., 0.8823.

5.4 Mechanism Analysis

In the following, we prove that TBUMA is truthful, individually rational, profitable, budget feasible, and computationally efficient.

Before showing the truthfulness, we first demonstrate that $\mathcal{V}(\boldsymbol{\omega})$ is monotone.

Lemma 1. *$\mathcal{V}(\boldsymbol{\omega})$ is monotone.*

Proof. According to (13), to show the monotonicity of $\mathcal{V}(\boldsymbol{\omega})$, it is sufficient to prove that $V^j(\boldsymbol{\omega})$ is monotone for any $j \in J$. Based on (19), we have

$$V^j(\boldsymbol{\omega} \cup \{i\}) - V^j(\boldsymbol{\omega}) = \big(\prod_{k=1}^{K} Q_j^k(\boldsymbol{\omega})\big)\big(1 - \prod_{k=1}^{K}(1 - q_{i,j}^k)\big) v_j^K + \sum_{k=1}^{K-1} \big(\prod_{m-1}^{k} Q_j^m(\boldsymbol{\omega})\big)\big(1 - \prod_{m=1}^{k}(1 - q_{i,j}^m)\big)\big(v_j^k - v_j^{k+1}\big). \quad (27)$$

Since $Q_j^k(\boldsymbol{\omega}) \in [0,1]$, $q_{i,j}^k \in [0,1]$ and $v_j^k - v_j^{k+1} \geq 0$, we conclude that $V^j(\boldsymbol{\omega} \cup \{i\}) - V^j(\boldsymbol{\omega}) \geq 0$, which completes the proof. □

Theorem 2. *TBUMA is truthful.*

Proof. The proof relies on the well-known Myerson's characterization [38]: a single-parameter mechanism is truthful iff (i) the winner selection rule is monotone: if bidder i wins by bidding b_i, it also wins by bidding $b_i' < b_i$ in a reverse auction; (ii) each winner is paid with the critical value, i.e., the highest bid price that makes it still win in a reverse auction. Since $\mathcal{V}(I)$ is independent of bid prices, to demonstrate truthfulness, it is sufficient to show these two properties under the case that $B < \mathcal{V}(I)$ and $B \geq \mathcal{V}(I)$, respectively. Let us study the case where $B < \mathcal{V}(I)$.

(*Monotonicity*): Recall that bidder w_n is the n-th winner in $\boldsymbol{\omega}$. By lowering b_{w_n} to b'_{w_n}, bidder w_n will become a candidate bidder for the k-th winner, where $k \leq n$ due to our greedy selection rule. Denote the alternate winner set as $\boldsymbol{\omega}'$ when bidder w_n lowers its price. To become the k-th winner, bidder w_n has to meet the budget feasible criterion. Since the first $k-1$ winners are the same for two cases, we have $\boldsymbol{\omega}'_{k-1} \subseteq \boldsymbol{\omega}_{n-1}$ ($\boldsymbol{\omega}'_{k-1}$ is the set of the first $k-1$ winners in $\boldsymbol{\omega}'$). According to the submodularity and monotonicity of $\mathcal{V}(\boldsymbol{\omega})$, we have

$$b'_{w_n} \leq b_{w_n} \leq \frac{B}{\min\{2, \frac{\mathcal{V}(I)}{B}\}} \cdot \frac{\mathcal{V}_{w_n}(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega}_{n-1} \cup \{w_n\})} \leq \frac{B}{\min\{2, \frac{\mathcal{V}(I)}{B}\}} \cdot \frac{\mathcal{V}_{w_n}(\boldsymbol{\omega}'_{k-1})}{\mathcal{V}(\boldsymbol{\omega}'_{k-1} \cup \{w_n\})}. \quad (28)$$

That is to say, bidder w_n still wins the auction by lowering its bid price. The monotonicity follows.

(*Critical value*): Considers the payment for winner $i \in \boldsymbol{\omega}$. According to (25), the payment is the maximum of $M+1$ values, where M is the total number of winners selected from $I_i = I \backslash \{i\}$. Let $r = \arg\max_{1 \leq m \leq M} \{\min\{\rho_m^i, \nu_m^i\}\}$ and $p_{(r)} = \min\{\rho_r^i, \nu_r^i\}$. We next show that bidding $b_i \geq p_{(r)}$ prevents bidder i from becoming the first M winners. Two cases are possible. If $p_{(r)} = \max_{1 \leq m \leq M} \rho_m^i$, it is clear that bidding a price higher than $p_{(r)}$ places bidder i after the first M winners. If $p_{(r)} < \max_{1 \leq m \leq M} \rho_m^i$, there should be some k that $p_{(r)} < \rho_k^i$. Due to the maximality of r, we have $\nu_k^i \leq p_{(r)} < \rho_k^i$, implying that bidding a price higher than $p_{(r)}$ violates the budget feasible criterion for becoming the k-th winner. For the remaining k that $p_{(r)} \geq \rho_m^k$, bidding a price higher than $p_{(r)}$ would place i after them. We therefore see that in both cases bidder i will be placed after the M winners. Furthermore, to be the $(M+1)$-th winner, there are no bidders that could win against i. Let $p_{(M+1)} = \min\{\mathcal{V}_i(\boldsymbol{\omega}_M^i), \nu_{M+1}^i\}$. Clearly, bidding a price higher than $p_{(M+1)}$ could prevent bidder i from being the $(M+1)$-th winner, since it results in either $b_i > V_i(\boldsymbol{\omega}_M^i)$ or $b_i > \nu_{M+1}^i$. On the other hand, if bidding $b_i = \max\{p_{(r)}, p_{(M+1)}\}$, bidder i would be either the r-th or $(M+1)$-th winner. In conclusion, $p_i = \max\{p_{(r)}, p_{(M+1)}\}$ is exactly the critical value for bidder i. Thus, our mechanism is truthful when $B < \mathcal{V}(I)$.

When $B \geq \mathcal{V}(I)$, the proof of truthfulness is similar to the proof above, and hence is omitted. In summary, our mechanism is truthful. □

Theorem 3. *TBUMA is individually rational.*

Proof. . Individual rationality means that $p_i - c_i \geq 0$ for $i \in I$. For bidder $i \in I \setminus \boldsymbol{\omega}$, the payoff is zero, and thus non-negative. Since the truthfulness has been proved, showing individual rationality is equivalent to demonstrating $p_i - b_i \geq 0$ for $i \in \boldsymbol{\omega}$. We only provide the proof for the case of $B \geq \mathcal{V}(I)$. For ease of presentation, we use i to denote the n-th winner w_n. Recall that w_n^i is the n-th winner in $\boldsymbol{\omega}^i$ (via running the winner selection for $I_i = I \backslash \{i\}$). The first $n-1$ winners are the same for both $\boldsymbol{\omega}$ and $\boldsymbol{\omega}^i$. If w_n^i exists, we have

$$\rho_n^i = \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1}^i) b_{w_n^i}}{\mathcal{V}_{w_n^i}(\boldsymbol{\omega}_{n-1}^i)} = \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1}) b_{w_n^i}}{\mathcal{V}_{w_n^i}(\boldsymbol{\omega}_{n-1})} \geq b_i. \quad (29)$$

The last inequality holds because bidder i is the n-th winner other than w_n^i. Based on the budget feasible criterion, we can get $\nu_n^i \geq b_i$, which follows $\min\{\rho_n^i, \nu_n^i\} \geq b_i$. Thus, we obtain $p_i \geq b_i$ from (25).

If w_n^i does not exist, bidder i must be the last winner in $\boldsymbol{\omega}$. We then have $\mathcal{V}_i(\boldsymbol{\omega}_{n-1}^i) = \mathcal{V}_i(\boldsymbol{\omega}_{n-1}) \geq b_i$ according to the winner selection procedure, and $b_i \leq \nu_n^i$ according to the budget feasible criterion, yielding $\min\{\mathcal{V}_i(\boldsymbol{\omega}_{n-1}^i), \nu_n^i\} \geq b_i$. According to (25), we still have $p_i \geq b_i$.

When $B \geq \mathcal{V}(I)$, the proof is similar to the above, and hence is omitted. Therefore, our mechanism is individually rational. □

Theorem 4. *TBUMA is profitable.*

Proof. Profitability means that the requester's utility is non-negative, i.e., $\mathcal{V}(\boldsymbol{\omega}) - \sum_{i \in \boldsymbol{\omega}} p_i = \sum_{1 \leq n \leq N} \mathcal{V}_{w_n}(\boldsymbol{\omega}_{n-1}) - \sum_{1 \leq n \leq N} p_{w_n} \geq 0$, where N is the total number of winners. Therefore, it is sufficient to prove that $p_{w_n} \leq \mathcal{V}_{w_n}(\boldsymbol{\omega}_{n-1})$ for each $1 \leq n \leq N$. For ease of presentation, we use bidder i to denote the n-th winner w_n. If $B < \mathcal{V}(I)$, let us consider the following two cases of p_i in (25).

If $p_i = \max_{1\le m\le M}\{\min\{\rho_m^i, \nu_m^i\}\}$, letting $r = \arg\max_{1\le m\le M}\{\min\{\rho_m^i, \nu_m^i\}\}$, we obtain

$$p_i = \min\{\rho_r^i, \nu_r^i\} \le \rho_r^i = \frac{\mathcal{V}_i(\boldsymbol{\omega}_{r-1}^i) b_{w_r^i}}{\mathcal{V}_{w_r^i}(\boldsymbol{\omega}_{r-1}^i)} \le \mathcal{V}_i(\boldsymbol{\omega}_{r-1}^i). \quad (30)$$

The last inequality is obtained via $V_{w_r^i}(\boldsymbol{\omega}_{r-1}^i) \ge b_{w_r^i}$. Since the first $n-1$ winners are the same for both $\boldsymbol{\omega}$ and $\boldsymbol{\omega}^i$, it follows that $p_i \le \mathcal{V}_i(\boldsymbol{\omega}_{r-1}^i) \le \mathcal{V}_i(\boldsymbol{\omega}_{n-1}^i) = \mathcal{V}_i(\boldsymbol{\omega}_{n-1})$, where the second inequality follows from $r \ge n$ and the submodularity.

Similarly, if $p_i = \min\{\mathcal{V}_i(\boldsymbol{\omega}_M^i), \nu_{M+1}^i\}$, we have $p_i \le \mathcal{V}_i(\boldsymbol{\omega}_M^i) \le \mathcal{V}_i(\boldsymbol{\omega}_{n-1}^i) = \mathcal{V}_i(\boldsymbol{\omega}_{n-1})$. Thus, the profitability follows when $B < \mathcal{V}(I)$.

If $B \ge \mathcal{V}(I)$, the proof is similar to the above, and hence is omitted. Thus, our mechanism is profitable. □

Before demonstrating the budget feasibility, we first prove the following lemma.

Lemma 2. *When $b_{w_n} \le B \cdot \frac{\mathcal{V}_{w_n}(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega}_{n-1}\cup\{w_n\})}$ is the budget feasible criterion for w_n, payment p_{w_n} is upper bounded by $\alpha \cdot B \cdot \frac{\mathcal{V}_{w_n}(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega})}$, where $\alpha = \min\{2, \frac{\mathcal{V}(I)}{B}\}$.*

Proof. We adapt the proof in [25] to prove this lemma, where $\alpha = 2$ in their case. We remark that the difference in α results from the stop criterion $\mathcal{V}_{i'}(\boldsymbol{\omega}_{n-1}) > b_{i'}$ in line 4 of Algorithm 2. For ease of presentation, we still use i to denote the n-th winner w_n. Suppose that by raising the bid price b_i to b_i', bidder i still wins and becomes the k-th winner of the alternate winner set $\boldsymbol{\omega}'$, where $n \le k$ due to the greedy selection rule. Since the payment is exactly the critical value, finding the upper bound of p_i is equivalent to finding the upper bound of b_i'. Letting $\boldsymbol{Q} = \boldsymbol{\omega} \setminus \boldsymbol{\omega}_k'$ ($\boldsymbol{\omega}_k'$ is the set of the first k winners in $\boldsymbol{\omega}'$), we use the case analysis to bound b_i'.

Case 1: If $\boldsymbol{Q} = \emptyset$, we have $\mathcal{V}(\boldsymbol{\omega}) \le \mathcal{V}(\boldsymbol{\omega}_k')$ according to the monotonicity of $\mathcal{V}(\boldsymbol{\omega})$. Therefore, we can obtain

$$\begin{aligned} b_i' &\le B \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{k-1}')}{\mathcal{V}(\boldsymbol{\omega}_k')} \le B \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1}')}{\mathcal{V}(\boldsymbol{\omega}_k')} \\ &= B \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega}_k')} \le B \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega})}, \end{aligned} \quad (31)$$

where the equality holds because the first $n-1$ elements of both winner sets are identical. By setting $b_i' = \alpha \cdot B \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega})}$, we obtain $\alpha \le 1$ in this case.

Case 2: If $\boldsymbol{Q} \ne \emptyset$, we have $b_i' \le B \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{k-1}')}{\mathcal{V}(\boldsymbol{\omega}_k')} \le B \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega}_k')}$. Still assuming $b_i' = \alpha \cdot B \cdot \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega})}$, we get

$$\alpha \le \frac{\mathcal{V}(\boldsymbol{\omega})}{\mathcal{V}(\boldsymbol{\omega}_k')}. \quad (32)$$

Moreover, consider adding a participant into $\boldsymbol{\omega}_k'$. For some participant r from set $\boldsymbol{Q}$, the marginal valuation per unit cost of r must be higher than that of the whole $\boldsymbol{Q}$:

$$\begin{aligned} \frac{\mathcal{V}(\boldsymbol{\omega}_k' \cup \boldsymbol{Q}) - \mathcal{V}(\boldsymbol{\omega}_k')}{\sum_{x\in\boldsymbol{Q}} b_x} &\le \frac{\mathcal{V}_r(\boldsymbol{\omega}_k')}{b_r} \le \frac{\mathcal{V}_r(\boldsymbol{\omega}_{k-1}')}{b_r} \\ \le \frac{\mathcal{V}_i(\boldsymbol{\omega}_{k-1}')}{b_i'} &\le \frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{b_i'} \le \frac{\mathcal{V}(\boldsymbol{\omega})}{\alpha \cdot B}. \end{aligned} \quad (33)$$

The third inequality is obtained from the fact that bidder i is the k-th winner instead of other bidders. Furthermore, it can be proven that $\sum_{i\in\boldsymbol{w}} b_i \le B$ (from Lemma 3.2 in [37]). Observing that $\sum_{i\in\boldsymbol{Q}} b_i \le \sum_{i\in\boldsymbol{w}} b_i \le B$ and $\mathcal{V}(\boldsymbol{\omega}_k' \cup \boldsymbol{Q}) \ge \mathcal{V}(\boldsymbol{\omega})$, we obtain

$$\frac{\mathcal{V}(\boldsymbol{\omega}) - \mathcal{V}(\boldsymbol{\omega}_k')}{B} \le \frac{\mathcal{V}(\boldsymbol{\omega})}{\alpha \cdot B}. \quad (34)$$

In addition, using the criterion $\mathcal{V}_i(\boldsymbol{\omega}_{k-1}') \ge b_i'$ (line 4 of Algorithm 2), we have

$$\alpha \le \frac{\mathcal{V}(\boldsymbol{\omega})}{B} \le \frac{\mathcal{V}(I)}{B}, \quad (35)$$

from (33). By combing (32), (35) and (34), we have $\alpha \le \min\{2, \frac{\mathcal{V}(I)}{B}\}$. Therefore, it follows that $\alpha \le \min\{2, \frac{\mathcal{V}(I)}{B}\}$ for both cases, which completes the proof. □

Theorem 5. *TBUMA is budget feasible.*

Proof. According to Lemma 2, when the budget feasible criterion $b_{w_n} \le B \cdot \frac{\mathcal{V}_{w_n}(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega}_{n-1}\cup\{w_n\})}$ is used, payment p_{w_n} is upper bounded by $\alpha \cdot B \cdot \frac{\mathcal{V}_{w_n}(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega})}$, where $\alpha = \min\{2, \frac{\mathcal{V}(I)}{B}\}$. Therefore, by adopting $b_{w_n} \le \frac{B}{\min\{2, \frac{\mathcal{V}(I)}{B}\}} \cdot \frac{\mathcal{V}_{w_n}(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega}_{n-1}\cup\{w_n\})}$ as the criterion instead, we have $\sum_{1\le n\le N} p_{w_n} \le \sum_{1\le n\le N} B \cdot \frac{\mathcal{V}_{w_n}(\boldsymbol{\omega}_{n-1})}{\mathcal{V}(\boldsymbol{\omega})} \le B$, where N is the number of winners.

However, the criterion only applies to the case of $B < \mathcal{V}(I)$. If $B \ge \mathcal{V}(I)$, we have $\sum_{i\in\boldsymbol{\omega}} p_i \le \mathcal{V}(\boldsymbol{\omega}) \le \mathcal{V}(I) \le B$, where the first inequality follows from the profitability. As a result, the budget feasibility can be naturally achieved when $B \ge \mathcal{V}(I)$. The proof is completed. □

Theorem 6. *TBUMA is computationally efficient.*

Proof. In each loop of Algorithm 2, we should calculate the value of $\frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{b_i}$. According to (13) and (20), we have $\mathcal{V}_i(\boldsymbol{\omega}_{n-1}) = \mathcal{V}(\boldsymbol{\omega}_{n-1}\cup\{i\}) - \mathcal{V}(\boldsymbol{\omega}_{n-1}) = \sum_{j\in J} V^j(\boldsymbol{\omega}_{n-1}\cup\{i\}) - V^j(\boldsymbol{\omega}_{n-1})$. Therefore, to obtain $\mathcal{V}_i(\boldsymbol{\omega}_{n-1})$, $V^j(\boldsymbol{\omega}_{n-1}\cup\{i\})$ and $V^j(\boldsymbol{\omega}_{n-1})$ should be computed $\mathcal{O}(|J|)$ times. The complexity of evaluating $V^j(\boldsymbol{\omega}_{n-1})$ is $\mathcal{O}(K)$ according to (11), where K is the subinterval number. Note that $Q_j^k(\boldsymbol{\omega})$ in (11) can be efficiently updated through $Q_j^k(\boldsymbol{\omega}_n) = (1 - q_{w_n,j}^k) Q_j^k(\boldsymbol{\omega}_{n-1})$ in each loop, with w_n being the n-th winner. Furthermore, finding the maximum $\frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{b_i}$ requires computing $\frac{\mathcal{V}_i(\boldsymbol{\omega}_{n-1})}{b_i}$ at most $|I|$ times, and the number of iterations for the while loop is upper bounded by $|I|$, yielding a total complexity of $\mathcal{O}(|I|^2|J|K)$.

For Algorithm 3, since the payment for each winner is computed by a process similar to Algorithm 2 and the number of winners is limited by $|I|$, the total running time is $O(|I|^3|J|K)$. The proof is completed. □

6 Performance Evaluation

To evaluate the performance of our proposed incentive mechanisms, we conduct trace-based simulations in this section.

6.1 Parameter Settings

Let us consider the use case in Section 1, where a police station intends to perform video analytics by leveraging vehicular crowdsourcing. In our simulations, wireless surveillance cameras are randomly distributed along the roadsides in an approximately 1.3km ×1.4 km area in Beijing, as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3052963, IEEE Transactions on Mobile Computing

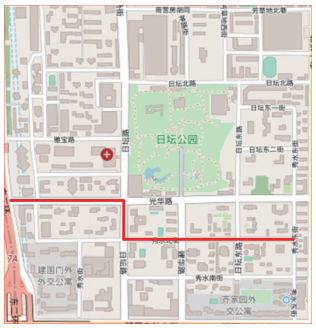


Fig. 5. The considered area in Beijing (latitude: from 39.9074 to 39.9192, longitude: from 116.4287 to 116.4444). The red line marks the route considered in Section 6.2 to evaluate our TTD estimation approach.

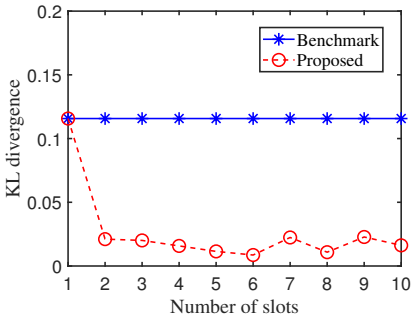


Fig. 6. Kullback-Leibler divergence between the estimated TTD and the ground truth TTD for a vehicle traversing a fixed path. A larger number of time slots corresponds to a finer discretization of the traffic conditions.

shown in Fig. 5. Without loss of generality, we set $D = 5$ minutes for the task valuation function in (1), where the $[0, D]$ is evenly partitioned into 5 subintervals, satisfying $v_j^1 = 1$, $v_j^2 = 0.8$, $v_j^3 = 0.6$, $v_j^4 = 0.4$, and $v_j^5 = 0.2$ for all task $j \in J$.

The trips of participating vehicles are generated by the widely-used traffic simulator, SUMO [39], based on the extracted road topology. Specifically, we use *RandomTrips* tool in SUMO to generate the trips. The arrivals of participating vehicles in the considered area follow a binomial distribution with the expected arrival rate λ ranging from 2/minute to 40/minute. Suppose the surveillance video has the resolution of 720×576 and frame rate of 30fps, and the adopted video analysis algorithm needs approximately 1 CPU cycle to process 1-bit data. We further assume that each surveillance video is 10 minutes. The computing workload of processing a 10-minute video (24bits for the RGB color values of a pixel) is about 179.2G cycles. Suppose that the available computing capability of vehicle $i \in I$, namely F_i, is uniformly drawn from the range $[10, 20]$GHz [10]. In our simulations, vehicle i sets $f_{i,j} = F_i$ for all $j \in J_i$, where $f_{i,j}$ is its CPU-cycle frequency allocated to task j. Due to the fact that the considered video chunk can be fast delivered to the passing vehicles by short-range transmission technique, we ignore the data transmission time in our simulation for simplicity[2]. In this way, the processing time $t_{i,j}^p$ in (3) can be obtained by $\frac{179.2GHz}{F_i}$. In addition, each bid is associated with a private cost satisfying $c_i = d_i + c_0|J_i|$, where c_0 is assumed to follow the uniform distribution over $[0.3, 1]$ [22], which accounts for the onboard resource consumption for each task; d_i is assumed to follow the uniform distribution over $[0.5, 1.5]$, which accounts for the drivers' efforts. The evaluation is conducted using MATLAB on a laptop with 2.81 GHz Intel(R) Core(TM) i7-7700HQ CPU and 8GB RAM.

6.2 Evaluation of TTD Estimation Approach

To proceed, we validate the effectiveness of the proposed TTD estimation approach. To reflect the time-varying traffic dynamics, we let the transportation network become increasingly congested. Specifically, we conduct the experiments for a 10-minute duration, where the transportation network is nearly free-flowing during the first two minutes, and after that, the vehicle arrival rate for the considered area increases to 60/ minute, which gradually saturates the road networks. During the offline link TTD estimation phase, we use 100 rounds of simulations to collect massive vehicle data to perform the fitting for each link TTD during the 10-minute duration. Then, at the 6-th minute, we pick up a vehicle departing on the trip with 6 links from the west to the east (the route is marked by a red line in Fig. 5) to test its trip time. The ground truth TTD for the trip time is obtained from 100 samples.

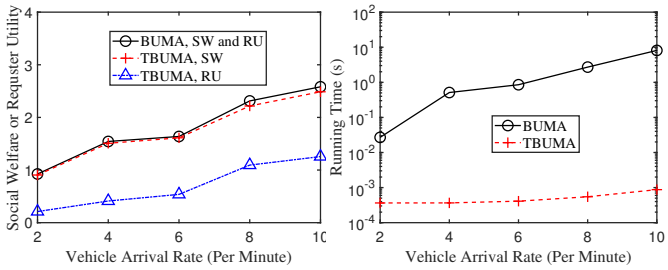
Fig. 6 uses Kullback-Leibler (KL) divergence as the performance metric to measure the similarity between the estimated TTD and the ground truth TTD versus the number of time slots that evenly divide the 10-minute period. Smaller KL divergence indicates a closer approximation. Since the ground truth TTD is a discrete function obtained from data, we also approximate our estimated TTD as a discrete function for comparison. The benchmark scheme estimates trip TTD by assuming that each link travel time follows a single normal distribution and summing them up. This method is equivalent to our approach with one time slot, which hence fails to capture the time-varying traffic conditions of each link. As a result, when the number of time slots is greater than 1, our approach leads to a significantly closer approximation to the exact TTD than the benchmark as observed from Fig. 6. Moreover, as the number of slots increases from 1 to 6, the KL divergence decreases due to the finer discretization of the traffic conditions. This curve begins to fluctuate when the slot number is larger than 7 because less samples are available within each slot for offline link TTD estimation.

6.3 Comparisons between BUMA and TBUMA

We evaluate the performance of TBUMA in terms of social welfare and the requester's utility by adopting BUMA as the benchmark. In the legends of the following figures, we use the abbreviations "SW" and "RU" to stand for "social welfare" and "requester's utility", respectively. For comparison with TBUMA, we assume $b_i = c_i$ for BUMA, even though it indeed fails to guarantee truthfulness. As illustrated in Fig. 7 (a), since the winners in BUMA are exactly paid

2. For example, according to [40], when H.264 is adopted for video compression, the transmission data size of the considered 10-minute video chunk can be estimated as 75MB, whereas IEEE 802.11ac can already achieve gigabit-per-second rate, making the transmission time negligible.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3052963, IEEE Transactions on Mobile Computing



(a) Requester's utility and social welfare.

(b) Running time.

Fig. 7. Comparison between BUMA and TBUMA. (a) Requester's utility and social welfare versus vehicle arrival rate λ, with $|J| = 20$ and $B = 15$. (b) Running time versus vehicle arrival rate λ, with $|J| = 20$ and $B = 15$.

with the bid prices, the social welfare is equivalent to the requester's utility. In contrast, TBUMA pays the winners with the payments that are potentially higher than their bid prices for ensuring the truthfulness, and therefore the requester's utility is lower than the social welfare due to the overpayment. Moreover, as validated by Fig. 7 (a), BUMA always outperforms TBUMA, as the subroutine $Greedy3$ of BUMA is always better than or equal to the greedy procedure adopted by TBUMA. Despite the superiority of BUMA, the social welfare achieved by TBUMA is indeed very close to that of BUMA. This result reveals that TBUMA achieves a good approximation performance. Besides, as shown in Fig. 7 (b), BUMA demands longer running time than TBUMA. For example, as the arrival rate goes to 10/minute, TBUMA uses about 0.001 second, whereas BUMA uses 8.11 seconds. Thus, in addition to achieving the truthfulness, TBUMA is also significantly superior to BUMA in terms of computational complexity.

6.4 Evaluation of TBUMA

In Fig. 8-10, we will evaluate the performance of TBUMA under the impact of various factors. For comparison, we employ the budgeted valuation maximization (BVM) mechanism as the benchmark, which refers to the mechanisms in [24], [25], [26] that aim to maximize the obtained task valuation (monotone submodular function) under budget limitation (they can be treated as one algorithm when adapted to our scenario). By comparing TBUMA with BVM in Fig. 8-10, we can see that TBUMA significantly outperforms BVM in terms of both social welfare and requester's utility when the budget is large (in Fig. 8), vehicle number is small (in Fig. 9), or task number is small (in Fig. 10). The reason is as follows. Unlike TBUMA that exploits the budget in a "conservative" way, BVM tends to recruit as many vehicles as possible within the budget limitation, implying that it may select vehicles whose marginal contributions on total task valuation do not deserve their true costs or received payments. As a consequence, the social welfare and requester's utility achieved by BVM diminish when the budget is huge or vehicle number is small, since the requester tends to utilize its budget to recruit vehicles that do not benefit the system (i.e., contributing a negative marginal social welfare/requester's utility). In particular, it is noteworthy the requester's utility of BVM can be negative in these figures, thus violating the critical property of profitability. On the other hand, it can be observed that BUMA and BVM achieve similar performance when the budget is small, vehicle number is great, or task number is great, since both algorithms rely on greedy selection rule and BVM does not tend to "waste" if the budget is tiny.

Fig. 11 illustrates the overpayment ratio versus the mean value of c_0, namely, $\overline{c}$, and the number of tasks, where c_0 is drawn from the interval $[\overline{c} - 0.3, \overline{c} + 0.3]$. The overpayment ratio is defined as $\gamma = \frac{\sum_{i \in \boldsymbol{\omega}} p_i - \sum_{i \in \boldsymbol{\omega}} c_i}{\sum_{i \in \boldsymbol{\omega}} c_i}$, which quantifies the extra cost used to ensure the truthfulness. From this figure, we observe that the overpayment ratio slightly increases with the budget. This is because ν_m^i in (23) increases with B, which results in the potential increase of the payment. More importantly, it is shown that the overpayment ratio keeps lower than 0.4 with varied parameters, revealing that guaranteeing truthfulness does not incur significantly extra payment in TBUMA.

Fig. 12 plots the budget utilization, i.e., $\frac{\sum_{i \in \boldsymbol{\omega}} p_i}{B}$, versus the given budget and the number of tasks. Since the budget utilization is lower than 1 as shown in Fig. 12, the budget feasibility is guaranteed. An interesting observation is that the budget utilization does not monotonically decrease with budget B. This is because more bidders are eligible to be selected when $B \geq \frac{\mathcal{V}(I)}{2}$ according to the budget feasible criterion in line 7 of Algorithm 2, thus leading to the improvement in budget utilization.

Fig. 13 and Fig. 14 show the individual rationality and truthfulness of TBUMA, respectively. Fig. 13 plots the empirical CDF of payoffs for all vehicles versus the given budget. Since no vehicle has a negative payoff, the individual rationality is demonstrated. In Fig. 14, we randomly select a winner to change its bid price in order to observe its received payment and payoff. The true cost and critical value of this winner are 1.30 and 1.78, respectively. As can be seen, the payment and payoff remain unchanged when the bid price is lower than the critical value, while reducing to zero when the bid price exceeds the critical value. This phenomenon reveals that the bidder cannot increase its payoff by lying about its true cost.

In Fig. 15, we demonstrate the timeliness-awareness of our incentive mechanism by showing the performance of TBUMA in handling tasks with diverse delay demands. For comparison, we consider another task valuation function satisfying that $V(t) = 0.5$ for $t \in [0, D]$, which means that the task valuation does not vary within the active interval. We call this new task valuation function as "function 2", while calling the task valuation function used before as "function 1". We vary deadline D from 200 seconds to 600 seconds. When employing function 1, the requester prefers vehicles providing shorter service delay. Instead, when employing function 2, the requester does not differentiate service delay within $[0, D]$, and therefore tends to select vehicles providing potentially long service delay yet with low bid prices. As such, function 2 induces less payment and longer average task delay than function 1, as demonstrated in Fig. 15. Furthermore, in both cases, the incurred payment decreases with D, whereas the average task delay increases with D. This is because when D is small, the requester has

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3052963, IEEE Transactions on Mobile Computing

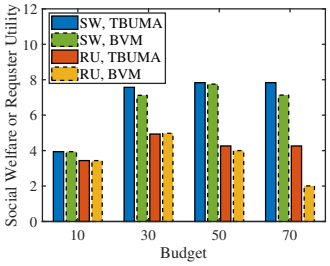


Fig. 8. Comparison between TBUMA and BVM versus budget B, with $\lambda = 16$/minute and $|J| = 60$.

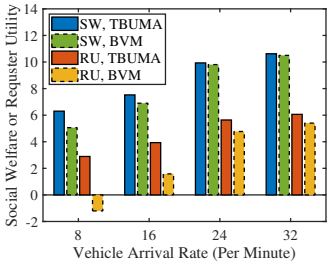


Fig. 9. Comparison between TBUMA and BVM versus vehicle arrival rate λ, with $|J| = 60$ and $B = 70$.

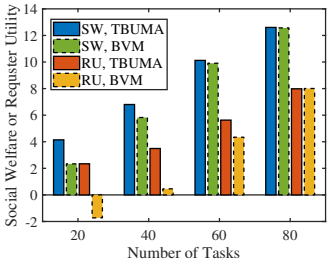


Fig. 10. Comparison between TBUMA and BVM versus number of tasks $|J|$, with $\lambda = 24$/minute and $B = 70$.

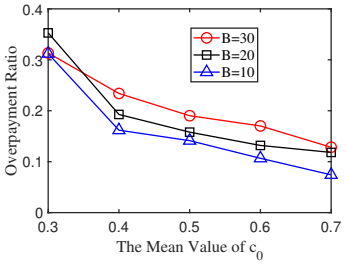


Fig. 11. Overpayment ratio versus the mean value of c_0, i.e., $\bar{c}$, and budget B, with $\lambda = 24$/minute and $|J| = 60$. c_0 is drawn from the interval $[\bar{c} - 0.3, \bar{c} + 0.3]$.

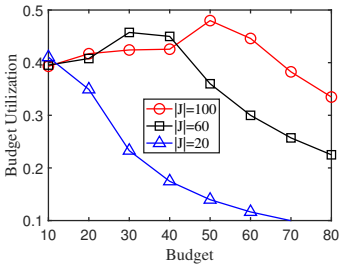


Fig. 12. Budget utilization versus budget B and number of tasks $|J|$, with $\lambda = 24$/minute.

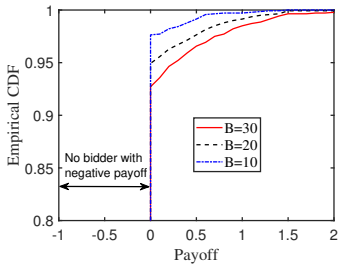


Fig. 13. Empirical CDF of payoffs for all participating vehicles versus budget B, with $\lambda = 8$/minute, and $|J| = 60$.

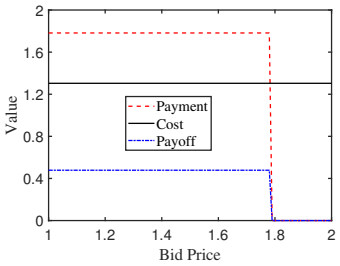


Fig. 14. Payoff and payment of a randomly selected participating vehicle versus its bid price.

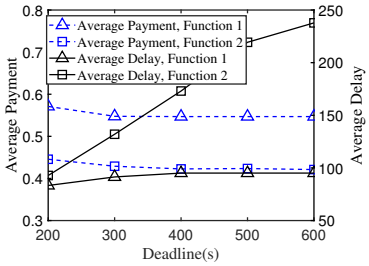


Fig. 15. Average payment (per completed task) and average task delay versus deadline D, with $\lambda = 24$/minute, $B = 30$ and $|J| = 60$.

to employ vehicles providing short service delay to meet the stringent deadline requirement, even at the cost of incurring more payment. The above observations indicate that our incentive mechanism can achieve the balance between task delay and incurred payment under different task valuation functions customized by requesters.

7 Conclusions

In this paper, we have proposed a reverse-auction-based incentive mechanism for vehicular crowdsourcing by explicitly taking the crucial factor, task delay, into consideration. By incorporating vehicle's travel time into task delay, we derive a tractable expression for the probability distribution of task delay based on a discrete-time traffic model. To maximize the utility of a service requester under a budget constraint, we cast the timeliness-aware winner selection and payment determination scheme as a non-monotone submodular maximization problem over a knapsack constraint. Then, we present TBUMA to solve this problem, which is shown to be truthful, budget feasible, profitable, individually rational and computationally efficient. Trace-based simulations have demonstrated the effectiveness of our proposed mechanisms in preserving nice economic properties and achieving delay-awareness simultaneously.

For the future work, we plan to study the task-partition model, where one task can be divided into several subtasks and be performed by multiple vehicles in parallel. This scenario has the requirement of integrity in the sense that one task is completed only if all the subtasks are completed, to which the current work cannot be simply extended. Another important research issue is to study the case where multiple service requesters co-exist and compete for vehicles' onboard resources.

References

[1] X. Chen, L. Zhang, B. Lin, and Y. Fang, "Delay-aware incentive mechanism for crowdsourcing with vehicles in smart cities," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, USA, December 2019.

[2] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The internet of things is the backbone," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, July 2016.

[3] H. Ding, C. Zhang, Y. Cai, and Y. Fang, "Smart cities on wheels: A newly emerging vehicular cognitive capability harvesting network for data transportation," *IEEE Wireless Communications*, vol. 25, no. 2, pp. 160–169, April 2018.

[4] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 146–152, June 2017.

[5] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, June 2016.

[6] H. Ding, Y. Fang, X. Huang, M. Pan, P. Li, and S. Glisic, "Cognitive capacity harvesting networks: Architectural evolution toward future cognitive radio networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1902–1923, Third Quarter 2017.

[7] H. Ding, X. Li, Y. Cai, B. Lorenzo, and Y. Fang, "Intelligent data transportation in smart cities: A spectrum-aware approach," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2598–2611, December 2018.

[8] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2982–2997, December 2018.

[9] P. Klemperer, "What really matters in auction design," *Journal of economic perspectives*, vol. 16, no. 1, pp. 169–189, March 2002.

[10] Y. Pang, L. Zhang, H. Ding, Y. Fang, and S. Chen, "Spath: Finding the safest walking path in smart cities," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 7071–7079, July 2019.

[11] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2542–2550.

[12] D. Han, W. Chen, and Y. Fang, "A dynamic pricing strategy for vehicle assisted mobile edge computing systems," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 420–423, April 2018.

[13] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Transactions on Vehicular Technology*, 2019.

[14] S. Liu, J. Tang, Z. Zhang, and J.-L. Gaudiot, "Computer architectures for autonomous driving," *Computer*, vol. 50, no. 8, pp. 18–25, August 2017.

[15] S. Zhou, Y. Sun, Z. Jiang, and Z. Niu, "Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 49–55, May 2019.

[16] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Reputation-aware, trajectory-based recruitment of smart vehicles for public sensing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1387–1400, May 2017.

[17] S. Xu, X. Chen, X. Pi, C. Joe-Wong, P. Zhang, and H. Y. Noh, "ilocus: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 8, pp. 1831–1847, August 2020.

[18] X. Peng, K. Ota, and M. Dong, "Multiattribute-based double auction toward resource allocation in vehicular fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3094–3103, April 2020.

[19] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, First Quarter 2016.

[20] M. H. Cheung, F. Hou, and J. Huang, "Delay-sensitive mobile crowdsensing: Algorithm design and economics," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2761–2774, December 2018.

[21] Y. Zhan, Y. Xia, Y. Liu, F. Li, and Y. Wang, "Incentive-aware time-sensitive data collection in mobile opportunistic crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 7849–7861, September 2017.

[22] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. ACM MobiCom*, Istanbul Turkey, August 2012, pp. 173–184.

[23] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *IEEE INFOCOM 2014*, Toronto, ON, Canada, May 2014, pp. 1231–1239.

[24] 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, July 2019.

[25] A. Singla and A. Krause, "Incentives for privacy tradeoff in community sensing," in *First AAAI Conference on Human Computation and Crowdsourcing*, Palm Springs, CA, USA, November 2013.

[26] Q. Zhang, Y. Wen, X. Tian, X. Gan, and X. Wang, "Incentivize crowd labeling under budget constraint," in *2015 IEEE INFOCOM*, Kowloon, Hong Kong, May 2015, pp. 2812–2820.

[27] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 9, pp. 1975–1989, 2019.

[28] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, Fourth Quarter 2017.

[29] M. Ramezani and N. Geroliminis, "On the estimation of arterial route travel time distribution with Markov chains," *Transportation Research Part B: Methodological*, vol. 46, no. 10, pp. 1576–1590, 2012.

[30] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen, "Estimating arterial traffic conditions using sparse probe data," in *13th international IEEE conference on intelligent transportation systems*. IEEE, 2010, pp. 929–936.

[31] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1679–1693, December 2012.

[32] K. Yin, W. Wang, X. B. Wang, and T. M. Adams, "Link travel time inference using entry/exit information of trips on a network," *Transportation Research Part B: Methodological*, vol. 80, pp. 303–321, 2015.

[33] V. Krishna, *Auction theory*. Academic press, 2009.

[34] A. Krause and D. Golovin, "Submodular function maximization." 2014.

[35] A. Gupta, A. Roth, G. Schoenebeck, and K. Talwar, "Constrained non-monotone submodular maximization: Offline and secretary algorithms," in *International Workshop on Internet and Network Economics*. Springer, 2010, pp. 246–257.

[36] U. Feige, V. S. Mirrokni, and J. Vondrák, "Maximizing non-monotone submodular functions," *SIAM Journal on Computing*, vol. 40, no. 4, pp. 1133–1153, July 2011.

[37] N. Chen, N. Gravin, and P. Lu, "On the approximability of budget feasible mechanisms," in *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, San Francisco, CA, USA, January 2011, pp. 685–699.

[38] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[39] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO-Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, December 2012.

[40] Fortinet, "Understanding IP surveillance camera bandwidth," *White Paper*, May 2017.

Xianhao Chen is currently pursuing the PhD degree at the University of Florida. His research interests include wireless networking, mobile crowdsourcing, and machine learning.

Lan Zhang is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Michigan Technological University. Her research interests include machine learning, wireless networking, cyber security for Internet-of-things and cyber-physical systems.

Yawei Pang is currently a Lecturer with the Department of IoT, School of Computer and Software, Nanjing University of Information Science and Technology. His research focuses on wireless networking, edge computing, and Internet of Things.

Bin Lin is currently a Full Professor with the Department of Information Science and Technology, Dalian Maritime University. Her current research interests include wireless communications, network dimensioning and optimization, resource allocation, artificial intelligence, maritime communication networks, edge/cloud computing, wireless sensor networks, and Internet of Things.

Yuguang Fang is currently a Distinguished Professor with the Department of Electrical and Computer Engineering, University of Florida. His research interests include wireless networks, privacy and security.