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/JIOT.2020.3044731, IEEE Internet of Things Journal

Incentivizing Platform–User Interactions for Crowdsensing

Chaocan Xiang, Suining He, Kang G. Shin, *Life Fellow, IEEE*, Yuben Qu, and Panlong Yang

Abstract—For effective crowdsensing, it is essential to incentivize the interactions of participants and platforms. Existing approaches do not tailor users' bidding to their preferences, *i.e., personalized bidding* (PB). To meet this need, we design an incentive mechanism, called *Picasso*, that achieves not only the expressiveness and description efficiency of PB for users, but also minimal social cost, computational efficiency, and strategy-proof for platform owners. This design is, however, challenging due to the intrinsic conflicting goals of the platform owner and users. To handle these conflicts, *Picasso* represents bids in a novel 3-D expression space by orchestrating three logical operations to balance among expressiveness, computational complexity, and description efficiency. Moreover, we equivalently decompose and recombine the complex task dependencies of bids originated from the expressiveness of PB, thus achieving a constant-factor approximation of optimal task allocation with strategy-proof in polynomial time. These properties of *Picasso* are proven theoretically. In addition to a detailed simulation study, our trace-driven evaluations show that, compared to existing approaches, *Picasso* can enable each user to bid 9.7x more tasks, on average, and decrease the description length by 74%, thus encouraging more users' participation. *Picasso* also reduces the platform owner's payment by more than 61%, hence yielding a win-win solution for incentivizing platform–user interactions.

Index Terms—Crowdsensing, Incentive Mechanism, Auction Model.

1 Introduction

The potential of crowds and pervasiveness of mobile devices have made crowdsensing increasingly popular and attractive, yielding numerous crowdsensing systems and platforms [21], [41], [43], such as Crowdsensing Map [44], Amazon Mechanical Turk, and Gigwalk[1]. The success of crowdsensing hinges on interactions between the platform (or the platform owner) and the crowdsensing participants [39]. It is, therefore, essential to incentivize the interactions of participants and platforms as witnessed from various proposals [22], [48], [51].

Of existing incentive mechanisms, the auction model has been widely studied as it increases competitiveness among bidding participants and incentivizes them better [12], [36]. Users bid for published tasks according to their preferences in describing bids, and then the platform uses certain criteria to allocate task(s) to each user. In practice, users have diverse preferences in bidding for a combination of tasks, owing to the differences in their in-situ context, interest, location, available time, *etc* [11]. For example, two users, Lucy and Bob, are interested in bidding for the same N tasks, but Lucy prefers only one of them due to her limited availability of time, while Bob with enough time wants any subset of these N tasks.

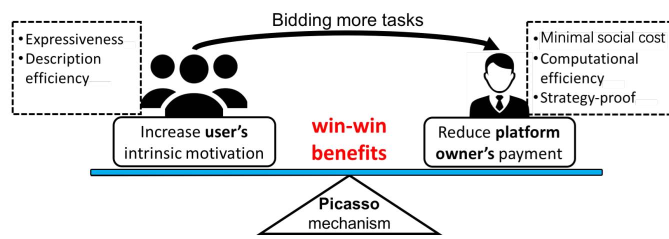


Fig. 1: Incentivization of platform–user interactions with win-win benefits using *Picasso*.

Enabling users' bidding tailored to their personal preferences, called *personalized bidding* (PB) [34], is key to incentivizing crowdsensing with benefits to both the participants and the platform owner. According to a recent survey [2] of over 1,500 users between the ages of 18 and 60, more than 56% of them prefer a service with personalized experience. In other words, PB can encourage the users to participate in crowdsensing by accommodating their personal preferences, raising the intrinsic motivation of psychological factors [9], [37]. Furthermore, users can be motivated to bid for more tasks for higher utility, promoting the competition among users, which, in return, reduces the platform's cost/payment.

However, prior work only focused on the design of task allocation with the minimum social cost, computational efficiency, and strategy-proof, all from the platform's perspective without considering the users' preference of PB [36]. On the one hand, most studies [27], [32], [49], [53] used single-minded bids, which cannot express the users' diverse preferences in PB [23]. For example, these single-minded bids, which let participants bid for either the bundle

- *Chaocan Xiang is with Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education, China; College of Computer Science, Chongqing University, 400044. E-mail: xiangchaocan@cqu.edu.cn.*
- *Suining He is with the Department of Computer Science and Engineering, the University of Connecticut, Storrs, CT, 06269. Email: suining.he@uconn.edu.*
- *Kang G. Shin is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48109-2121. Email: kgshin@umich.edu.*
- *Yuben Qu is with Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Email: quyuben@sjtu.edu.cn.*
- *Panlong Yang is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China, 230026. E-mail: plyang@ustc.edu.cn.*

1. http://www.gigwalk.com/

of all tasks or nothing [24], [49], do not allow Lucy to express her preference for a combination of tasks. Thus, they cannot support the *expressiveness* of PB, *i.e.,* the ability that a mechanism allows users to express all possible task combinations in their bids. On the other hand, a few researchers [15], [23], [50] have recently considered the users' expressiveness by using multi-minded bids. Nevertheless, they do not satisfy expressiveness or description efficiency of PB from the user's perspective.

To fill this gap, as illustrated in Fig. 1, we propose a comprehensive incentive mechanism that achieves (i) *expressiveness* and (ii) *description efficiency* of PB in describing users' bids; (iii) *minimal social cost*, (iv) *computational efficiency*, and (v) *strategy-proof* in the platform owner's allocation of tasks to the users [15], [50]. However, it is very challenging to achieve all of these simultaneously due to the intrinsically conflicting goals of the platform owner and participants:

- **Chg1. Describing bids**: PB's expressiveness enables a much larger space of candidate task allocations due to the increasing number of task bids (*i.e.,* task bundles), at the expense of significantly higher computational complexity for the optimal task allocation. Moreover, the required bidding flexibility via PB's expressiveness lengthens the users' bid descriptions, rendering them inefficient to use. Hence, it is difficult to achieve expressiveness without degrading description efficiency.
- **Chg2. Allocating tasks**: various bidding options owing to expressiveness add more complex constraints upon the allocation of different tasks to users, called *task dependency*, making it difficult to solve the task-allocation problem for minimizing social cost. This problem is proved to be NP-hard in Sec. 3.4. Moreover, such dependency can be abused by selfish users to strategically misreport and manipulate for higher utility, making the task allocation less strategy-proof.

To address these two challenges, we propose a novel incentive mechanism for crowdsensing, called ***Picasso***[2]. As shown in Fig. 1, in contrast to prior work, *Picasso* achieves all five features (i)-(v) from the perspectives of both the platform and users in the following two key steps.

To address Chg1, in Sec. 4.1, we build a formal framework of bid description in 3-D expressive space by combining three logical operations, *i.e.,* AND, XOR, and OR. Further, based on this framework, a new PB description method is proposed to achieve an excellent balance among expressiveness, computational complexity, and description efficiency.

To address Chg2, in Sec. 4.2, we first construct a task dependency graph to model the dependencies of task allocations in a user's PB. Then, by jointly considering the relationships among the logical operations, we decompose the complex graph of a user's PB into multiple subgraphs of independent single-minded bids for more tractable task allocation. Moreover, we recombine such subgraphs of a user to design an adaptive critical-payment computation scheme, preventing users' strategic exploitation of task dependencies for high utility. Finally, the above properties of *Picasso* are evaluated via theoretical analyses in Sec. 4 and trace-based Gigwalk case-studies in Sec. 5.

In summary, this paper makes three main contributions:

- Design of a comprehensive framework for describing users' bids and generalizing prior work. Based on this framework, a new PB description method is devised by leveraging a 3-D expression space with orchestration of AND, XOR, and OR, achieving a good trade-off among expressiveness, computational complexity, and description efficiency.
- Design of a dependency-aware task allocation algorithm, achieving constant-factor approximation and strategy-proof in polynomial time by decomposing and then re-combining the task dependency graph.
- Extensive theoretical analyses and trace-driven Gigwalk case-studies to evaluate the performance of *Picasso*. Our trace-driven evaluations show that unlike existing approaches [15], [27], [50], *Picasso* enables each user to bid for 9.7x more tasks, on average, and decrease the description length by 74%, encouraging more user participation. As a result, it reduces the social cost and the platform's payment by more than 60% and 61%, respectively. That is, *Picasso* provides a win-win solution to incentivize interactions between the users and the platform owner, promoting long-term crowdsensing.

The rest of this paper is organized as follows. First, we discuss the related work in Sec. 2, then state the system model and formalize the problem in Sec. 3. We also propose an incentive mechanism called *Picasso* along with theoretical analyses in Sec. 4. In Sec. 5, we conduct traces-driven evaluations, followed by discussing influenced factors in Sec. 6 and concluding remarks in Sec. 7.

2 Related Work

Overview: There have been numerous studies of incentivized crowdsensing [47], [51], most of which use the reverse auction model [13], [36]. As *Picasso* falls into this category, we focus on reviewing its related studies in terms of users' bids, classified as *single-minded* and *multi-minded bids* [19], [30], [50]. Other orthogonal studies, such as the posted-pricing model [20], [35], can be found in [36], [51].

Single-minded bids: from the *platform*'s perspective, most existing studies are based on single-minded bids (*SMB*) due to ease of design, and focus on task allocation. For example, many of them aim at maximizing the platform's profit [32] or achieving constant-factor approximation [27] subject to other constraints, such as computational efficiency [7], [53], strategy-proof [38], [49], quality constraint [27], budget limitation [53], or social network [29], [31], [40], [46]. Despite their reported promising results for the platform, all of them are based on single-minded bids, *i.e.,* each user is allowed to select one bundle of tasks in a 'win all or nothing' fashion [19]. Thus, they fail to consider users' diverse preferences for task bundling [42]. Without diversified bid design [19], they cannot satisfy the design goal of expressiveness, thus discouraging users' participation by decreasing psychological intrinsic motivations [9], [17]. In contrast, from the users' perspectives, *Picasso* caters for the user's diverse bids in terms of their social and psychological

2. Like the Cubist painting pioneered by Pablo *Picasso*, we describe the bids in 3-D space by decomposing and recombining the graph.

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/JIOT.2020.3044731, IEEE Internet of Things Journal

TABLE 1: Comparison between our work and existing works from the perspectives of platform and users.

Researches		User's Perspective		Platform's Perspective		
References	Bid's type	Expressiveness	Description efficiency	Guaranteed near-minimal social cost	Computational efficiency	Strategy proof
Yang *et al.* [24], [49]	Single-mind	×	✓	×	✓	✓
Jin *et al.* [26], [27]	Single-mind	×	✓	✓	×	✓
Tang *et al.* [25], [38]	Single-mind	×	✓	✓	✓	✓
Feng *et al.* [15]	Multi-mind	×	✓	✓	✓	✓
Jin *et al.* [23]	Multi-mind	✓	×	✓	✓	×
Zhang *et al.* [50]	Multi-mind	✓	×	×	✓	✓
Our work	**Multi-mind**	✓	✓	✓	✓	✓

differences [37]. *Picasso* is complementary to these state-of-the-arts, and can further incentivize the users with both extrinsic and intrinsic motivations [9].

Multi-minded bids: Recently, a few researchers [15], [19], [23], [30], [50] considered *multi-minded bids* in designing incentive mechanisms from the perspective of users. Han *et al.* [19] focused on the posted-pricing model, which is bid-independent. Hence, it is inapplicable to our scenarios based on the bid-dependent auction model [19]. Lin *et al.* [30] focused on the protection of user privacy and security attack, which is orthogonal to our work.

One line of prior works [15], [23], [50], highly related to this paper, investigates the design of incentive mechanisms with multi-minded bids in the auction model. Feng *et al.* [15] designed the TRAC mechanism, where each user can submit multiple disjoint bids, and get any subset of them. We generalize it to the Single-OR-Bidding (*SOB*) model, which is proved to be inexpressive (Sec. 4.1.2). Although QoI-MRC [23] and IMC-SM [50] can satisfy the user's expressiveness, they neglect the impact of multi-minded bids on the allocation of tasks and payments. For example, QoI-MRC is untruthful for payment allocation, while IMC-SM cannot achieve guaranteed near-optimal social cost in task allocation. Moreover, their designs [23], [50], as a special case of the Single-XOR-Bidding (*SXB*) model, are proved to be description-inefficient with exponential length. In contrast, we build a generic framework of bid description which easily accommodates these results [15], [23], [50]. Based on this framework, we design a novel bid description scheme, decreasing the description length to polynomial complexity. Moreover, we propose a new task allocation algorithm, achieving constant-factor sub-optimization and truthfulness in polynomial time.

In addition, researchers [5], [6], [28] studied the bidding language of combinatorial auction, based on which bidders express their complex preferences on bundles of expected tasks. They focus on either the expressiveness to exhaustively elicits user's preferences [28] or the description efficiency which promotes user-friendliness and communication [6]. Instead of studying the bidding language independently, our work jointly considers the relationship between its design and task allocation in crowdsensing scenarios. It is fed back to refine the bid description scheme to balance the features of platform and users.

Summary: as summarized in Table 1, in comparison with existing approaches, *Picasso* is a novel incentive mechanism from the perspectives of both the platform and users fulfilling all the five important features. With such 'win-win' benefits, *Picasso* augments the platform–user interaction and incentivizes the entire crowdsensing, promoting long-term development of the crowdsensing community [48].

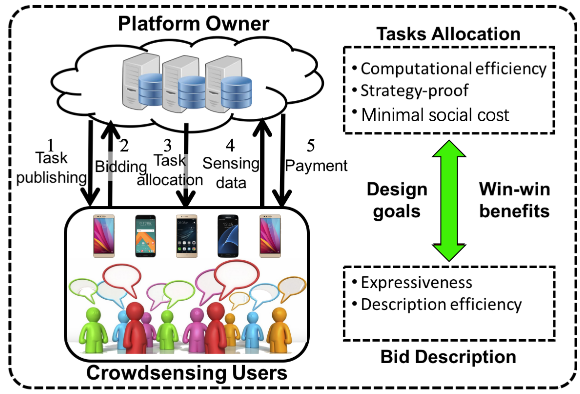


Fig. 2: System model of incentive mechanism with design goals.

3 System Model & Problem Formulation

We first introduce the system model of the auction-based incentive mechanism, followed by giving a toy example of personalized bidding in Gigwalk. We then formulate the mechanism design problem with perspectives of both users and platform owner. Finally, this problem is theoretically proved to be NP-hard.

3.1 System Model

Fig. 2 illustrates the general model for the incentive mechanism of crowdsensing, whose workflow consists of the following three phases.

Step 1 (Task publishing): The platform publishes sensing tasks to the crowd of users who might be interested in this crowdsensing campaign. Let $\mathcal{T}$ be the set of tasks, *i.e.*, $\mathcal{T} = \{\tau_j | j \in \{1, \dots, M\}\}$, where τ_j and M denote the j-th task and the number of tasks, respectively. For each τ_j, there is a corresponding valuation $v_j > 0$. Assume there are N users interested in the tasks, and let $\mathcal{U} = \{u_i | i \in \{1, \dots, N\}\}$ denote the set of users.

Step 2 (Task bidding): According to the description method, each user u_i makes her/his PB for those tasks based on their preference r_i, including the tasks they want to perform and the desired payments. We define the PB of u_i as $B_i = \{b_{i,k} | b_{i,k} = (T_{i,k}, a_{i,k}), k \in \{1, \dots, \delta_i\}\}$, where $b_{i,k}$ and δ_i denote the atomic bid of u_i as Def. 1 and the number of atomic bids, respectively. Last, they send the PBs to the platform.

Definition 1. ***Atomic bid****, also called single-minded bid (SMB): a user can submit a bid $b_{i,k} = (T_{i,k}, a_{i,k})$, where $T_{i,k}$ is a subset of tasks (i.e., $T_{i,k} \subseteq \mathcal{T}$) and $a_{i,k}$ is the desired payment for executing these tasks. The user may execute all the tasks in $T_{i,k}$ with the payment $a_{i,k}$, or not execute any task with no payment. Such a bid is said to be atomic, i.e., the basic unit of bid description.*

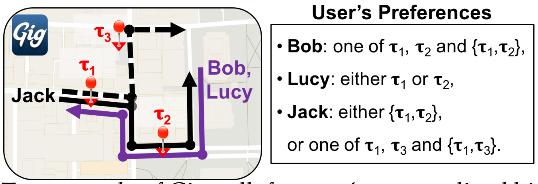


Fig. 3: Toy example of Gigwalk for user's personalized bidding, where τ_j denotes the j-th task.

In addition to the desired payment $a_{i,k}$, u_i incurs a real cost $c_{i,k}$ of executing $T_{i,k}$, which is in practice private information and only known to herself/himself. Due to the human's selfishness and rationality, users prefer not to ask for their real costs in the bids so as to earn more. Hence $c_{i,k} \leq a_{i,k}$.

Steps 3, 4 & 5 (Allocation, execution, and payment of tasks): based on their bids (*i.e.,* $\{B_i|i = 1, \ldots, N\}$), the platform determines the set of allocated tasks $\mathcal{S}_i$ and the payment p_i for u_i, according to the task allocation and the payment rules (Step 3). Note that $\mathcal{S}_i = \bigcup_{k=1}^{\delta_i} \mathcal{S}_{i,k}$, where $\mathcal{S}_{i,k}$ denotes the set of task allocation for $T_{i,k}$, *i.e.,* $\mathcal{S}_{i,k} \subseteq T_{i,k}$. Let $p_i = \sum_{k=1}^{\delta_i} p_{i,k}$, where $p_{i,k}$ denotes the payment of the task allocation $\mathcal{S}_{i,k}$, depending on its desired payment $a_{i,k}$. Each u_i then executes the tasks (*i.e.,* $\mathcal{S}_i$) assigned to her/him by sensing, and reports the sensing results to the platform (Step 4), which then pays p_i to u_i (Step 5). We assume that all the tasks can be executed successfully, thanks to a large number of potential users with diverse skills in crowdsensing [15]. We also assume that the users can successfully finish their allocated tasks, while discussing the users' unreliability in Sec. 6.

Hence, the **utility of user** u_i is $\pi_i^{\mathrm{u}} = \sum_{k=1}^{\delta_i}(p_{i,k} - c_{i,k})$. Furthermore, all the users finish the set of tasks as $\mathcal{S} = \bigcup_{i=1}^{N} \mathcal{S}_i$, and the **utility of platform** is $\pi^{\mathrm{p}} = \sum_{\forall \tau_j \in \mathcal{S}}(v_j - c_j)$. Table 2 illustrates frequently used notations.

3.2 Example of Personalized Bidding Scenario

We take Gigwalk as an example to illustrate the PB in crowdsensing. Gigwalk is a widely deployed crowdsensing app. Its platform enables mobile participants to visit shops at different locations to collect real-time data about specific products, as shown in Fig. 3.

Suppose Gigwalk publishes three sensing tasks (*i.e.,* τ_1, τ_2, and τ_3) at three different locations, and has three participants (*i.e.,* Bob, Lucy, and Jack). Due to the differences in their interests, contexts, and availabilities, these three users have different preferences on task bidding. Specifically, both Bob and Lucy will go through the locations of τ_1 and τ_2 along with the purple line routine in Fig. 3. As Bob has enough time and expects to do any subset of τ_1 and τ_2 with the prices \$50 and \$10, respectively. However, Lucy has limited time and expects to bid either τ_1 or τ_2 with the prices \$10 and \$30, respectively. Jack has two alternative routines as the black solid line and the black dashed line in Fig. 3. He wants to either do the bundle of τ_1 and τ_2 in one routine with prices \$15 and \$35, respectively, or take either τ_1 or τ_3 for the price of \$15 or \$10, respectively.

Let u_1, u_2, and u_3 denote Jack, Bob, and Lucy, respectively. Then, according to the system model in Sec. 3.1, the personalized bids of Jack, Bob, and Lucy are formalized as $B_1 = \{(\tau_1), (\tau_3), (\tau_1, \tau_2), (\tau_1, \tau_3)\}$, $B_2 = \{(\tau_1), (\tau_2), (\tau_1, \tau_2)\}$, and $B_3 = \{(\tau_1), (\tau_2)\}$, respectively.

TABLE 2: Frequently Used Notations

Symbols	Definitions
$\tau_j, c_j, M, \mathcal{T}$	j-th task; its cost; its number; set of tasks.
$u_i, N, \mathcal{U}$	i-th user; its number; set of users.
$B_i, b_{i,k}$	u_i's bid; k-th atomic bid of u_i, *i.e.,* $b_{i,k} \in B_i$.
$T_{i,k}, a_{i,k}, p_{i,k}$	task set of $b_{i,k}$; its bidding price; its payment.
$\mathcal{S}_i, u_{i,k}^{\mathrm{v}},$	set of allocated tasks for u_i; k-th virtual user of u_i.
E, ξ	expressive power; cost-efficiency.
λ, π	description length; utility of user(platform).

3.3 Problem Formalization

As shown in Fig. 2, the mechanism design problem for personalized bidding can be stated as:

1) from the perspective of the platform, how to design the *task allocation algorithm* $\Psi(\cdot)$ for the platform to allocate all the tasks $\mathcal{T}$ to the users $\mathcal{U}$ with the payments based on the users' PBs ($\mathbf{B} = \{B_1, \ldots, B_N\}$) as Eq. (3), so as to achieve (i) *minimal social cost*, (ii) *strategy-proof*, and (iii) *computational efficiency*.
2) from the perspective of users, how to design a *bid description method* $\Omega(\cdot)$ for u_i to describe PBs B_i based on her/his preferences r_i as Eq. (2), in order to satisfy (iv) *expressiveness* and (v) *description efficiency*;

Let $x_{i,j}$ be the indicator variable ($x_{i,j} \in \{0, 1\}$), *i.e.,* $x_{i,j} = 1$ if task j is allocated to u_i, and $x_{i,j} = 0$ otherwise. $\mathbf{X}_i = \{x_{i,j}|j = 1, \cdots, M\}$. The problem is formulated as:

$$\textbf{Min} \quad \sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j}c_{i,j} \tag{1}$$

$$\textbf{s.t.} \quad B_i = \Omega(r_i), i \in \{1, \ldots, N\}, \tag{2}$$

$$(\mathbf{X}_i, p_i) = \Psi(\mathbf{B}, \mathcal{T}, \mathcal{U}), i \in \{1, \ldots, N\}, \tag{3}$$

$$\sum_{j=1}^{M}(p_{i,j} - c_{i,j}) \geq 0, \forall i \in \{1, \ldots, N\}, \tag{4}$$

$$\pi_i^{\mathrm{u}}(c_i, a_{-i}) \geq \pi_i^{\mathrm{u}}(a_i, a_{-i}), \forall i \in \{1, \ldots, N\}, \tag{5}$$

$$\sum_{i=1}^{N} x_{i,j} = 1, \forall j \in \{1, \ldots, M\}, \tag{6}$$

where the features of the platform owner and users are:
(i) **Minimal social cost**: from the perspective of the platform, it aims at maximizing the platform utility, *i.e.,*

$$\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j}(v_j - c_{i,j}). \tag{7}$$

As shown in Eq. (6), each task is constrained to be allocated to at most one user, and all the tasks should be completed [49]. Hence, $\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j}v_j$ is a constant. The objective function can be equivalently represented as Eq. (1), *i.e.,* minimizing the *social cost* (or called *social welfare* [38]), which is the sum of the user's real costs of all tasks s/he finished [15].
(ii) **Strategy-proof**: the mechanism should have *individual rationality* [27], *i.e.,* all the users receive non-negative utilities as Eq. (4). Also, it should satisfy *truthfulness* [53] as Eq. (5), meaning that it is a dominant strategy in a Nash equilibrium for all the users to claim the real costs c_i in their bids, 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/JIOT.2020.3044731, IEEE Internet of Things Journal

a_i denotes the bidding price of u_i, and a_{-i} denotes those of the other users. As each user cannot improve her/his utility by misreporting the costs individually, it makes the mechanism truthful. This is based on the assumption that the users are independent and will not collude with each other [52]. In addition, we only consider the tasks covered by bids of at least two users for truthfulness, due to the large number of potential users with diverse skills in crowdsensing [15].
(iii) **Computational efficiency**: an algorithm is computationally efficient if and only if it can be completed in polynomial time [53]. In the PB formulation, the task allocation algorithm $\Psi(\cdot)$ should be computationally efficient for real-time allocation, which is very important for incentivizing users in practice [19]. To simplify the formulation, as Eq. (1), we consider the additive cost/payments of tasks, where the total cost/payments of multiple tasks are the summation of that of each individual task [50] [51]. Also, we will discuss the non-additive cost of tasks [10] in Sec. 6.
(iv) **Expressiveness**: the description method $\Omega(\cdot)$ should be flexible enough to allow the users to express their diverse bidding preferences of task combinations [33]. The set of all possible task combinations in users' bids allowed by $\Omega(\cdot)$ is referred to as its *expressive space* [28]. The size of expressive space is then defined as *expressive power* (denoted as E) which quantifies the expressiveness of a description method [33]. The larger the expressive power is, the more diversities of preferences the users can express.
(v) **Description efficiency**: (also referred to as *Description succinctness*). The bid description method $\Omega(\cdot)$ should be efficient for users to express their preferences [28]. The number of atomic bids in a bid description is defined as its description length (λ) [34]. We also use *average description length* (ADL) $\overline{\lambda}$ of all the descriptions to quantify the description efficiency of $\Omega(\cdot)$. Intuitively, a shorter ADL eases bidding description for participants, leading to higher description efficiency [34]. Moreover, the computational complexity of $\Omega(\cdot)$ is dominated by the maximum description length (MDL) $\widehat{\lambda}$ of all the descriptions.

3.4 Analysis of Problem Complexity

Given the above comprehensive problem formalization for both the platform and users, we analyze the computational complexity of task allocation with personalized bidding.

Theorem 1. *The optimal task allocation problem with PB is NP-hard.*

Proof. Recall that B_i is the PB of u_i. Let $b_{i,k} := (T_{i,k}, a_{i,k})$ be a single-minded bid by u_i as Def. 1. Then, we have $b_{i,k} \in B_i$. If we replace B_i by $b_{i,k}$ for each $u_i \in \mathcal{U}$, constraint in Eq. (2) can be relaxed and the optimal task allocation problem with PB (called OTA-PB) becomes the one without PB (named as OTA-NonPB).

We next demonstrate the NP-hardness of OTA-PB by proving that OTA-NonPB is at least NP-hard. Let $\mathcal{S}_i$ be the set of sensing tasks assigned to u_i to execute and c_i be the corresponding total cost. Thus, in problem OTA-NonPB, constraint in Eq. (3) is equal to a set cover constraint over task set $\mathcal{T}$, *i.e.*, $\bigcup_{i=1,2,\ldots,N} \mathcal{S}_i = \mathcal{T}$, meaning that all sensing tasks will be executed.

Furthermore, constraint in Eq. (6) implies that the intersection of any two different sets $\mathcal{S}_i$ and $\mathcal{S}_j$ ($\forall i, j \in \{1, 2, \ldots, N\}, i \neq j$) is null, *i.e.*, $\mathcal{S}_i \bigcap \mathcal{S}_j = \emptyset$. As the strategy-proof is decided by not the task allocation but the payment, constraints in Eqs. (4) and (5) can be relaxed [36]. As a result, OTA-NonPB will become:

$$\text{Min} \quad \sum_{i=1}^{N} c_i \tag{8}$$

$$\text{s.t.} \quad \bigcup_{i\in\{1,\ldots,N\}} \mathcal{S}_i = \mathcal{T}, \tag{9}$$

$$\mathcal{S}_i \bigcap \mathcal{S}_j = \emptyset, \forall i, j \in \{1, 2, ..., N\}, i \neq j. \tag{10}$$

The decision version of the above problem is a minimum weighted set cover (MWSC) problem with the mutual exclusiveness constraint as Eq. (10) [18]. Note that MWSC is a well-known NP-complete problem [3]. Since checking whether an obtained solution satisfies the mutual exclusiveness constraint or not could be completed in polynomial time, the decision problem belongs to NP [3]. Therefore, the OTA-NonPB is NP-hard, which establishes the NP-hardness of OTA-PB. □

4 Design of *Picasso*

To solve the mechanism design problem with PB in Sec. 3.3, as illustrated in Fig. 4, we propose *Picasso*, which efficiently allocates the tasks with the truthful payment to the users according to their diverse preferences on the published tasks. Specifically, *Picasso* consists of the following two main components:

1) **Bid description based on 3-D space** (Sec. 4.1): we first propose a formal framework of bid description based on 3-D expressive space, created by the orchestration of AND, XOR, and OR in Sec. 4.1.1. Then, in Sec. 4.1.2, it is proved theoretically to balance among expressiveness, description efficiency, and computational complexity, via comparison with existing models.
2) **Task allocation based on dependency graph** (Sec. 4.2): we first build the task dependency graph model to represent the user's bid in Sec. 4.2.1. Then, in Sec. 4.2.2, we design the task allocation scheme based on graph decomposition to address the NP-hard problem. It achieves a near-optimal solution with a guaranteed approximation ratio in polynomial time cost. Finally, in Sec. 4.2.3, we propose a novel payment method based on graph recombination, leveraging the critical payment computation to devise the strategy-proof payment scheme. Such scheme can prevent selfish users from strategically exploiting the complex PB to improve their utilities.

4.1 Bid Description in 3-D Space

4.1.1 PB Description in 3-D Space

We first build a formal framework for the bid description using 3-D expressive space, and then propose a specific description method along with a walk-through example.

(1) Formal framework of bid description using 3-D expressive space.

To satisfy the expressiveness, description efficiency, and computational efficiency, we leverage three basic logical

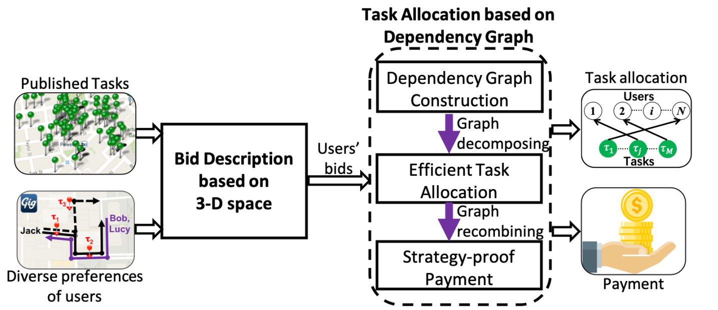


Fig. 4: Framework of *Picasso*, incentivizing platform–user interactions.

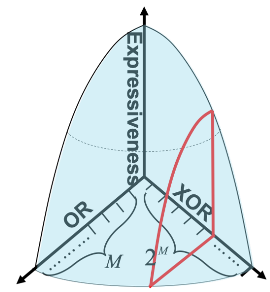


Fig. 5: PB description via 3-D space.

operators, *i.e.*, AND, XOR, and OR, to describe users' bids as Def. 2.

Definition 2. ***XOR-of-OR Bidding Description****: it is constructed based on* AND*,* XOR*, and* OR *in the following three steps.*
(1) ***Construct atomic bids****: each u_i can submit an atomic bid, denoted by $b_{i,k}$, including an arbitrary number (e.g., $H_{i,k}$) of task pairs $(\tau_{i,k,h}, a_{i,k,h})$ by* AND *($\wedge$), $h \in \{1,\ldots,H_{i,k}\}$. It implies that the user expects to be allocated all of the tasks $T_{i,k} = \{\tau_{i,k,h} | h \in \{1,\ldots,H_{i,k}\}\}$ with the total payment $a_{i,k} = \sum_{h=1}^{H_{i,k}} a_{i,k,h}$, or none of the tasks with no payment. Thus, $b_{i,k} = (T_{i,k}, a_{i,k})$.*
(2) ***Construct* OR *bids****: each u_i can submit an* OR *bid, denoted by b_i^{O}, which includes an arbitrary number (e.g., K_i) of disjoint atomic bids $b_{i,k}$ by the logical operator* OR *($\cup$), i.e., $b_i^{\mathrm{O}} = \bigcup_{k=1}^{K_i} b_{i,k}$. $\forall k_1, k_2 \in \{1,\ldots,K_i\}$ and $k_1 \neq k_2$, we have $T_{i,k_1} \cap T_{i,k_2} = \emptyset$. This implies that the user expects to be allocated the tasks of any subset of these atomic bids with the sum of their respective payments.*
(3) ***Construct XOR-of-OR bids****: each u_i can submit an* XOR*-of-*OR *bid denoted by b_i^{XO}, including an arbitrary number (e.g., L_i) of* OR *bids $b_{i,l}^{\mathrm{O}}$ by the logical operator* XOR *($\oplus$), i.e., $b_i^{\mathrm{XO}} = \bigoplus_{l=1}^{L_i} b_{i,l}^{\mathrm{O}} = \bigoplus_{l=1}^{L_i} \bigcup_{k=1}^{K_{i,l}} b_{i,l,k}$. It implies that the user expects allocation of at most one of these* OR *bids, e.g., $b_{i,l}^{\mathrm{O}}$.*

Theorem 2. XOR*-of-*OR *bidding description in terms of x* XOR *and y* OR *operators has the expressive power $E^{\mathrm{XO}}(x,y)$ as Eq. (11) with $\widehat{\lambda} = \mathcal{O}(x \cdot y)$. Moreover, it can represent all the PBs with the largest number of* XOR*s and* OR*s (i.e., $x = 2^M$ and $y = M$).*

$$E^{\mathrm{XO}}(x,y) = \sum_{i=1}^{x} \frac{E^{\mathrm{O}}(y)!}{i!(E^{\mathrm{O}}(y)-i)!}, \tag{11}$$

where the expressive power $E^{\mathrm{O}}(y)$ with y OR operators is $E^{\mathrm{O}}(y) = \sum_{k=1}^{y} \sum_{i=1}^{k} \frac{(-1)^i (k-i)^M}{i!(k-i)!}$. Note that $x \in \{1,\ldots,E^{\mathrm{O}}(y)\}$, and $y \in \{1,\ldots,M\}$.

Proof. We prove this theorem by using dynamic programming and the Inclusion-Exclusion Principle theorem of combinatorics [18]. See Appendix for a detailed proof. □

According to Eq. (11), the increment of $E^{\mathrm{XO}}(x,y)$ *w.r.t.* x is

$$\Delta_x E^{\mathrm{XO}}(x,y) = \frac{E^{\mathrm{O}}(y)!}{x!(E^{\mathrm{O}}(y)-x)!}. \tag{12}$$

According to Eqs. (11) and (12), we can create a 3-D expressive space as shown in Fig. 5. In this 3-D space, the x-axis and y-axis represent the number of XORs (*i.e.*, x) and ORs (*i.e.*, y), respectively. The z-axis represents the increase of expressive space by adding the x-th XORs with y ORs as $\Delta_x E^{\mathrm{XO}}(x,y)$. Thus, we can use this 3-D space to represent the expressiveness of XOR-of-OR bidding description method. Moreover, according to Theorem 2, this 3-D expressive space with the largest number of XORs and ORs is equivalent to that of PB. Thus, we can use AND, OR, and XOR to describe all PBs in a 3-D space.

(2) PB description method: we use the above XOR-of-OR bidding framework to describe PBs. In order to trade off between expressiveness and computational complexity, we reduce the 3-D expressive space by limiting the length of one dimension (*e.g.*, XOR or OR) with a constant R. According to Eq. (11), the dominant dimension of entire expressive space for computational complexity is the number of XORs. Thus, we further constrain the number of XORs by a constant R, and the expressive power is given by Eq. (13) with $\widehat{\lambda} = R \cdot M$.

$$E^{\mathrm{XO}}(R) = \sum_{i=1}^{R} \frac{E^{\mathrm{O}}(M)!}{i!(E^{\mathrm{O}}(M)-i)!}, \quad R \in \{1,\ldots,E^{\mathrm{O}}(M)\}. \tag{13}$$

Hence, the PB description method consists of the following steps.

1) *Describe a bundle of tasks:* if a user expects to be allocated a bundle of tasks, s/he creates an atomic bid for this bundle of tasks. Otherwise, s/he creates an atomic bid for each task.
2) *Describe union of tasks:* if the user expects to be allocated any subset of the tasks, s/he uses OR based on the atomic bids to create the plan.
3) *Generate R exclusive plans:* based on the above two steps, each user can iteratively create exclusive plans with the maximum limit R. Each participant uses XOR to describe it, and expects to be allocated tasks of at most one of these plans.

Let us consider the Gigwalk example in Sec. 3.2 to show how to describe the PB based on 3-D space. Jack has two exclusive plans and uses XOR to describe his PB as Eq. (14). Similarly, the PBs of Bob and Lucy are given by Eqs. (15) and (16), respectively. In addition, we will discuss how to enable user-friendly PB description in Sec. 6.

$$\text{Jack}: \quad \{(\tau_1 \wedge \tau_2, \$50)\} \oplus \{(\tau_1, \$15) \cup (\tau_3, \$10)\}, \tag{14}$$
$$\text{Bob}: \quad \{(\tau_1, \$50) \cup (\tau_2, \$10)\}, \tag{15}$$
$$\text{Lucy}: \quad \{(\tau_1, \$10) \oplus (\tau_2, \$30)\}. \tag{16}$$

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/JIOT.2020.3044731, IEEE Internet of Things Journal

4.1.2 Theoretical Analysis

Based on the framework in Sec. 4.1.1, we first present the models of *SOB* and *SXB* as Defs. 3 and 4, respectively, generalizing the multi-minded bids [15], [23], [50]. We then compare ours with *SMB*, *SOB*, and *SXB* in terms of the expressiveness and the description efficiency via theoretical analysis.

Definition 3. ***SOB Bidding Description***: *it is constructed using operators* AND *and* OR *in two steps:*
(1) construct atomic bids: same as in Def. 2.
(2) construct OR *bids: same as in Def. 2.*

Definition 4. ***SXB Bidding Description***: *it is constructed using operators* AND *and* XOR *in two steps:*
(1) construct atomic bids: same as in Def. 2.
(2) construct XOR *bids: each u_i can submit an* XOR *bid denoted by b_i^{X}, which includes an arbitrary number (e.g., K_i) of atomic bids $b_{i,k}$ by* XOR *operations ($\oplus$), i.e., $b_i^{\mathrm{X}} = \bigoplus_{k=1}^{K_i} b_{i,k}$. It implies that the user expects allocation of at most one of these atomic bids, e.g., getting the set of tasks $T_{i,k}$ of $b_{i,k}$ with payment $a_{i,k}$.*

According to Defs. 3 and 4, we have the following propositions.

Proposition 1. *SOB bidding description has the expressive power $E^{\mathrm{O}}(x)$ which is formally given by*

$$E^{\mathrm{O}}(x) = \sum_{k=1}^{x}\sum_{i=1}^{k} \frac{(-1)^i (k-i)^M}{i!(k-i)!}, \quad x \in \{1, \ldots, M\}, \quad (17)$$

with the MDL $\widehat{\lambda} = x$, when the number of OR *operations is x. However, it cannot represent all the PBs with the largest number of* OR*s.*

Proof. We prove that $E^{\mathrm{O}}(x)$ is equal to Eq. (17) by using dynamic programming and the Inclusion-Exclusion Principle theorem of combinatorics. Then, we exploit the *reductio ad absurdum* method to prove the *SOB* bidding description cannot represent all PBs. In fact, it only represents one kind of bids with no substitutability [34]. See Appendix for a detailed proof. □

Proposition 2. *SXB bidding description has the expressive power $E^{\mathrm{X}}(x)$ as Eq. (18) with the MDL $\widehat{\lambda} = x$, when the number of* XOR *operators is x. Moreover, SXB has the same expressive power as* Picasso*, and both of them can represent all the PBs with the largest number of* XOR*s.*

$$E^{\mathrm{X}}(x) = \sum_{i=1}^{x} \frac{(2^M)!}{i!(2^M - i)!}, \quad x \in \{1, \ldots, 2^M\}. \quad (18)$$

Proof. We prove this using Def. 4 and a detailed proof is provided in Appendix. □

Proposition 3. *Given M tasks, ADL of* Picasso *is $\mathcal{O}(M)$ that is on the same scale as SMB and SOB, while that of SXB is $\mathcal{O}(2^M)$ for the same expressive power.*

Proof. We prove it based on Eqs. (11), (17), and (18), and a detailed proof is provided in Appendix. □

Summary: according to Props. 1, 2, and 3, both *SMB* and *SOB* are inexpressive, which cannot accommodate all the expressive space of PB. Although *SXB* can satisfy the expressiveness, it is not description-efficient with ADL $\mathcal{O}(2^M)$. In contrast, *Picasso* achieves a better trade-off between the expressiveness and description efficiency than XOR, *i.e.*, to achieve the same expressive power, *Picasso* reduces ADL from $\mathcal{O}(2^M)$ to $\mathcal{O}(M)$.

4.2 Task Allocation Based on Dependency Graph

4.2.1 Construction of Task Dependency Graph

According to the formal framework of bid description in Sec. 4.1, u_i's PB can be formally described as

$$b_i^{\mathrm{XO}} = \bigoplus_{l=1}^{L_i} \bigcup_{k=1}^{K_{i,l}} \bigwedge_{h=1}^{H_{i,l,k}} (\tau_{i,l,k,h},\ a_{i,l,k,h}), \quad (19)$$

where $(\tau_{i,l,k,h},\ a_{i,l,k,h})$ denotes a task. $\wedge$, $\cup$, and $\oplus$ represent 3 different dependencies between task allocation, *i.e.*, AND, OR, and XOR, respectively. For example, in Jack's PB description as Eq. (14), τ_1 and τ_2 have AND dependency and should be allocated together; τ_1 and τ_3 have OR dependency, and any subset of them can be allocated. $(\tau_1 \wedge \tau_2)$ and $(\tau_1 \cup \tau_3)$ have XOR dependency, and at most one of them can be assigned. Thus, the PB description of a user consists of many different kinds of complex task dependencies.

As illustrated in Fig. 6b, we use a graph $G = (\mathcal{T}, \mathbf{e})$, called *task dependency graph*, to represent the PB description in Fig. 6a. Specifically, the vertices represent tasks $\tau_j, j \in \{1, \ldots, M\}$. The edge $e = (\tau_j, \tau_{j'}) \in \mathcal{T} \times \mathcal{T}$ represents the allocation dependency between τ_j and $\tau_{j'}$. They include AND, OR, and XOR dependencies, which are represented by the purple, green, and red edges, respectively, in Fig. 6b.

Using this task dependency graph, we propose a dependency-aware task allocation and adaptive critical-payment computation method by *decomposing* and then *recombining* the task dependency graph of PB.

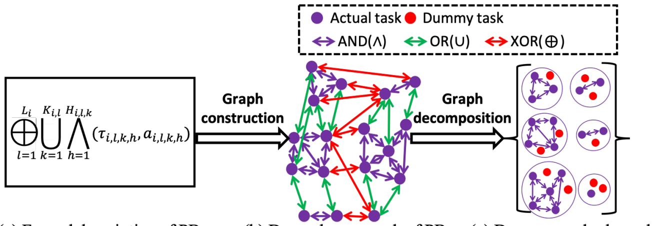
4.2.2 PB Decomposition for Efficient Task Allocation

The task dependency graph is usually complex due to PB description, making the direct allocation rather difficult. To address this, we transform this complex problem with PB description into a simple problem with independent SMB ones by decomposing the task dependency graph. We then propose a greedy-based allocation algorithm to achieve constant-factor approximation with polynomial time cost for this NP-hard problem.

(1) Problem transformation by decomposing the task dependency graph: we leverage the properties of the logical operators and their intrinsic relationships to decompose the XOR and OR dependencies for the problem transformation.

First, as shown in the red circles in Fig. 6b, we use *dummy tasks* d with no intrinsic values and costs to express XOR constraints indirectly, decomposing the XOR dependencies. It is because the XOR dependency between the task sets T_i and $T_{i'}$ is equivalent to the OR dependency by adding a dummy task $d_{i,i'}$ for each of them, *i.e.*, $T_i \oplus T_{i'} \iff (T_i \wedge d_{i,i'}) \cup (T_{i'} \wedge d_{i,i'})$, where $d_{i,i'}$ represents the dummy task which is added for T_i and $T_{i'}$. Specifically, as shown in Lines 2–3 of Alg. 1, for $b_{i,l,k}$ in u_i's PB, we add a dummy task $d_{l,k,l',k'}$ for each $b_{i,l',k'}$ inside different OR bids with $b_{i,l,k}$, *i.e.*, $l' \neq l$.

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/JIOT.2020.3044731, IEEE Internet of Things Journal



(a) Formal description of PB (b) Dependency graph of PB (c) Decomposed subgraphs

Fig. 6: Construction, decomposition, and recombination of task dependency graph based on PB model.

Thus, as shown in Fig. 6b, by adding the dummy tasks to decompose these XOR dependencies, we transform the XOR-of-OR bidding description in Eq. (19) to that with *SOB* bidding description as $\bigcup_{k=1}^{K_{i,l}} \bigwedge_{h=1}^{H_{i,l,k}} (\tau_{i,l,k,h} \wedge d_{i,l,k,h}, a_{i,l,k,h})$.

Furthermore, we transform the PB description with OR to that with independent SMBs by decomposing OR dependencies. We add *virtual users* with SMBs to represent OR dependency. Specifically, as shown in Line 4 of Alg. 1, for each $b_{i,l,k}$ of u_i, we create a virtual user $u^{\mathrm{v}}_{i,l,k}$ with an SMB $b_{i,l,k}$, where $b_{i,l,k} = \bigwedge_{h=1}^{H_{i,l,k}} (\tau_{i,l,k,h} \wedge d_{i,l,k,h}, a_{i,l,k,h})$.

Let δ_i be the number of virtual users for u_i, *i.e.*, $\delta_i = K_{i,l}$. Such transformation is to leverage the similarity of properties between the OR dependency and the task allocation. According to Def. 2, atomic bids of an OR bid have disjunction and independent properties. In other words, like the task allocation of different users, the disjoint atomic bids for each virtual user can be independently assigned to this user.

In summary, based on the above decomposition of XOR and OR dependencies, the task allocation problem with a complex PB description as Eq. (19) is equivalently transformed to the simple problem only with SMBs.

Proposition 4. *A user's PB description of length λ can be equivalently transformed to λ independent SMB bids of λ virtual users by adding at most λ^2 dummy tasks.*

Proof. See Appendix for the proof. □

(2) Task allocation with constant-factor approximation: based on the above transformation, the optimal task allocation problem with PB is transformed to the one with SMB bids. This problem is proved to be NP-hard in Sec. 3.4. Thus, we propose an approximate task allocation scheme by greedily selecting the virtual users $u^{\mathrm{v}}_{i^*,k^*}$ who are the most cost-efficient as

$$u^{\mathrm{v}}_{i^*,k^*} = \operatorname*{arg\,max}_{\forall u^{\mathrm{v}}_{i,k} \in \mathcal{U}^{\mathrm{v}}} (\xi_{i,k} | T_{i,k} \cap \mathcal{S} = \emptyset), \tag{20}$$

where $T_{i,k}$ denotes the task set of the virtual user $u^{\mathrm{v}}_{i,k}$. $\mathcal{U}^{\mathrm{v}}$ denotes the set of unselected virtual users. $\mathcal{S}$ represents the set of selected tasks. $\xi_{i,k}$ denotes the cost efficiency of $u^{\mathrm{v}}_{i,k}$ with SMB bid $(T_{i,k}, a_{i,k})$, *i.e.*, $\xi_{i,k} = \frac{\sqrt{|T_{i,k}|}}{a_{i,k}}$, where $|T_{i,k}|$ is the number of tasks in $T_{i,k}$, and the dummy tasks (*e.g.*, $d_{i,k}$) contribute 0.

Specifically, as in Line 7 of Alg. 1, we first sort all the virtual users (*e.g.*, $u^{\mathrm{v}}_{i,k}$) according to the decreasing cost efficiency $\xi_{i,k}$. Then, we iteratively select the most *cost-effective* virtual user $u^{\mathrm{v}}_{i,k}$ whose bidding task set $T_{i,k}$ is disjoint with the set of the allocated tasks $\mathcal{S}$, until all the tasks are allocated, as illustrated in Lines 8-15 of Alg. 1. Note that $\mathbf{B}^{\mathrm{s}}$ in Line 11 of Alg. 1 denotes the set of selected atomic bids.

Algorithm 1 : Task & Payment Allocation in *Picasso*

Input: Task set: $\mathcal{T} = \{\tau_1, \tau_2 \cdots, \tau_M\}$; Bid set of users: $\{b^{\mathrm{XO}}_i | b^{\mathrm{XO}}_i = \bigoplus_{l=1}^{L_i} \bigcup_{k=1}^{K_{i,l}} \bigwedge_{h=1}^{H_{i,l,k}} (\tau_{i,l,k,h},\ a_{i,l,k,h}), i \in [1,N]\}$;

Output: Task&payment allocation of users: $\{(\mathcal{S}_i, p_i), i \in [1,N]\}$;

1: %*Equivalent Decomposing of Task Dependency Graph*
2: **while** $(\forall i \in \{1,\ldots,N\}, \forall b_{i,l,k} \in B_i)$ **do**
3: $\quad \forall b_{i,l',k'} \in B_i$ and $l' \neq l$, Create $d_{l,k,l',k'}$, $T_{i,l,k} = T_{i,l,k} \wedge \{d_{l,k,l',k'}\}$;
4: $\quad$ Create $u^{\mathrm{v}}_{i,l,k}$ with $b_{i,l,k} = (T_{i,l,k}, a_{i,l,k})$;
5: **end while**
6: %*Greedy Allocation of Tasks based on Cost Efficiency*
7: With $\xi_{i,k}$, sort $u^{\mathrm{v}}_{i,k}$ $(\forall i \in \{1,\ldots,N\}, \forall k \in \{1,\ldots,\delta_i\})$ in descending order as $\mathcal{U}^{\mathrm{v}}$. Let $\mathcal{S} = \emptyset, \mathbf{B}^{\mathrm{s}} = \emptyset$;
8: **while** $(\mathcal{T} \backslash \mathcal{S} \neq \emptyset)$ **do**
9: $\quad$ Let $u^{\mathrm{v}}_{i^*,k^*}$ denote the first user of $\mathcal{U}^{\mathrm{v}}$;
10: $\quad$ **if** $(T_{i^*,k^*} \cap \mathcal{S} \neq \emptyset)$ **then**
11: $\quad\quad \mathcal{U}^{\mathrm{v}} = \mathcal{U}^{\mathrm{v}} \backslash \{u^{\mathrm{v}}_{i^*,k^*}\}$;
12: $\quad$ **else**
13: $\quad\quad \mathcal{S} = \mathcal{S} \cup T_{i^*,k^*}$, $\mathbf{B}^{\mathrm{s}} = \mathbf{B}^{\mathrm{s}} \cup b_{i^*,k^*}$;
14: $\quad$ **end if**
15: **end while**
16: %*Strategy-proof Payment Allocation*
17: **while** $(\forall i \in \{1,\ldots,N\}, \forall k \in \{1,\ldots,\delta_i\})$ **do**
18: $\quad$ **if** $(b_{i,k} \in \mathbf{B}^{\mathrm{s}})$ **then**
19: $\quad\quad$ Compute $p_{i,k}$ according to Eqs. (23) and (24);
20: $\quad$ **else**
21: $\quad\quad p_{i,k} = 0$;
22: $\quad$ **end if**
23: **end while**
24: **return** $\mathcal{S}_i = \bigcup_{\forall b_{i,k} \in \mathbf{B}^{\mathrm{s}}} T_{i,k}$, $p_i = \sum_{k=1}^{\delta_i} p_{i,k}$, $\forall i \in \{1,\ldots,N\}$.

(3) Theoretical analysis: we analyze the proposed task allocation scheme in terms of the approximation ratio and the computing complexity as follows.

Lemma 1. *Alg. 1 solves the problem with a constant factor $\sqrt{M}$ of the optimal solution, given M tasks.*

Proof. Let $\mathbf{B}^{\mathrm{s}}$ and $\mathbf{B}^*$ be the set of selected atomic bids for *Picasso* and the optimal solutions, respectively. For $\forall b_k \in \mathbf{B}^*$, we create $\mathbf{B}^{\mathrm{s}}_k = \{b_i \in \mathbf{B}^{\mathrm{s}} | \xi_i \geq \xi_k, T_i \cap T_k \neq \emptyset\}$. As $c_i \leq c_k \cdot \sqrt{|T_i|}/\sqrt{|T_k|}$, we have $\sum_{b_i \in \mathbf{B}^{\mathrm{s}}_k} c_i \leq \frac{c_k}{\sqrt{|T_k|}} \sum_{b_i \in \mathbf{B}^{\mathrm{s}}_k} \sqrt{|T_i|}$. Using the Cauchy-Schwarz inequality, we have $\sum_{b_i \in \mathbf{B}^{\mathrm{s}}_k} \sqrt{|T_i|} \leq \sqrt{|\mathbf{B}^{\mathrm{s}}_k|} \sqrt{\sum_{b_i \in \mathbf{B}^{\mathrm{s}}_k} |T_i|}$. As $\forall b_i \in \mathbf{B}^{\mathrm{s}}_k$, $T_i \cap T_k \neq \emptyset$ and $\forall b_{i_1}, b_{i_2} \in \mathbf{B}^{\mathrm{s}}_k$, $T_{i_1} \cap T_{i_2} = \emptyset$, $|\mathbf{B}^{\mathrm{s}}_k| \leq |T_k|$. Moreover, $\sum_{b_i \in \mathbf{B}^{\mathrm{s}}_k} |T_i| \leq M$. Hence, based on the above derivations, we have

$$\sum\nolimits_{b_i \in \mathbf{B}^{\mathrm{s}}_k} c_i \leq \sqrt{M} \cdot c_k. \tag{21}$$

We define $\mathbf{B}^{\mathrm{s}*} = \bigcup_{\forall b_k \in \mathbf{B}^*} \mathbf{B}^{\mathrm{s}}_k$. Then, according to Eq. (21), $\sum_{b_i \in \mathbf{B}^{\mathrm{s}*}} c_i \leq \sqrt{M} \cdot \sum_{b_k \in \mathbf{B}^*} c_k$. Since $\mathbf{B}^{\mathrm{s}} \subseteq \mathbf{B}^{\mathrm{s}*}$, $\sum_{b_i \in \mathbf{B}^{\mathrm{s}}} c_i \leq \sum_{b_i \in \mathbf{B}^{\mathrm{s}*}} c_i$. Finally, we have

$$\sum\nolimits_{b_i \in \mathbf{B}^{\mathrm{s}}} c_i \leq \sqrt{M} \cdot \sum\nolimits_{b_k \in \mathbf{B}^*} c_k, \tag{22}$$

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/JIOT.2020.3044731, IEEE Internet of Things Journal

where $\sum_{b_i \in \mathbf{B}^s} c_i$ and $\sum_{b_k \in \mathbf{B}^*} c_k$ denote the social cost achieved by *Picasso* and the optimal solution, respectively. Thus, Lemma 1 holds.

□

Lemma 2. *Given M tasks and N users, the time complexity of Alg. 1 is $\mathcal{O}(M^2N^2)$, while those of SMB, SOB, and SXB are $\mathcal{O}(N^2)$, $\mathcal{O}(M^2N^2)$, and $\mathcal{O}(N^2)$, respectively.*

Proof. See Appendix for the proof.

Based on Lemmas 1 and 2, we finally have Theorem 3.

Theorem 3. *Picasso achieves computational efficiency and approximates the optimal solution with a constant factor $\sqrt{M}$.*

4.2.3 PB Recombination for Strategy-proof Payment

In Sec. 4.2.2, by decomposing the complex PB into independent SMBs, we transform the problem to a form with efficient task allocation. Given this transformation, we first design the truthful payment scheme based on the critical value for independent SMBs without considering PB, *i.e.,* non-PB. We then show the PBs make the user's bids more complex, thus leading to the untruthfulness issue for the payment scheme. To address this new issue, we also design the payment scheme for PB based on graph recombination, which is finally proved to have truthfulness and individual rationality.

(1) Truthful payment scheme for non-PB based on critical prices: according to the Truthful Theorem [34], the auction-based mechanisms on single parameter domain are truthful if and only if the following two conditions hold:

- **monotonicity**: the task allocation scheme is monotone. Specifically, for u_i, if the bid $b_i = (T_i, a_i)$ is selected for task allocation, then her/his bid $\widetilde{b}_i = (T_i, a_i - \delta)$ is still selected when $\delta > 0$.
- **critical price**: each user should be paid the critical price for her/his selected bid. The critical price p_i is the minimum one for u_i, such that her/his bid (T_i, a_i) would not be selected if $a_i > p_i$.

As we use the task allocation scheme based on greedy selection in Sec. 4.2.2, it satisfies the monotone condition for truthfulness. Thus, in order to hold the truthful property, we utilize the critical payment to compute remittance of each virtual user for her/his SMB.

Specifically, as in Lines 17–23 of Alg. 1, the virtual users without task allocation get no payment. On the other hand, according to the bid $(T_{\hat{i}^*,\hat{k}^*}, a_{\hat{i}^*,\hat{k}^*})$ of the critical user $u^{\text{v}}_{\hat{i}^*,\hat{k}^*}$, the selected virtual user $u^{\text{v}}_{i,k}$ with $b_{i,k} = (T_{i,k}, a_{i,k})$ gets the payment as

$$p_{i,k} = a_{\hat{i}^*,\hat{k}^*} \cdot \frac{\sqrt{|T_{i,k}|}}{\sqrt{|T_{\hat{i}^*,\hat{k}^*}|}}, \tag{23}$$

where $u^{\text{v}}_{\hat{i}^*,\hat{k}^*} = \arg\max_{u^{\text{v}}_{\hat{i},\hat{k}}} \{\xi_{\hat{i},\hat{k}} | T_{\hat{i},\hat{k}} \cap T_{i,k} \neq \emptyset, \xi_{\hat{i},\hat{k}} \neq \xi_{i,k}\}$.

(2) Truthful payment scheme for PB based on graph recombination: although the above mechanism based on the critical payment guarantees the users to be truthful in terms of the SMB bidding, it does not work for users' PBs. Taking Jack in Fig. 3 as an example, we assume that $u^{\text{v}}_{1,2}$ is selected, and $u^{\text{v}}_{1,1}$ is the critical user of $u^{\text{v}}_{1,2}$. According to Eq. (23), the payment of Jack is $a_{1,1} \cdot \sqrt{|T_{1,2}|}/\sqrt{|T_{1,1}|}$. Thus, Jack can strategically misreport $a_{1,1}$ to improve his own total utility. The reasons are as follows. A user with PB can be decomposed into multiple virtual users with SMB bids. An individual virtual user cannot directly improve his own utility (*i.e.,* earnings) by misreporting, he may strategically help other virtual users improve their respective utilities, thus enhancing the total utility of that actual user. As a result, *the PB can be strategically utilized by the selfish users to improve their utilities, hence inducing untruthfulness*.

To address such untruthfulness, we recombine the task dependency graph of PB and design an adaptive critical-payment computation. Specifically, for each selected user, we find a critical user from the group of different actual users who have intersecting (common) tasks with the selected one.Formally, for the k-th virtual user of u_i, *i.e.,* $u^{\text{v}}_{i,k}$, we find the critical user $u^{\text{v}}_{\hat{i}^*,\hat{k}^*}$ as

$$u^{\text{v}}_{\hat{i}^*,\hat{k}^*} = \arg\max_{u^{\text{v}}_{\hat{i},\hat{k}}} \{\xi_{\hat{i},\hat{k}} | \hat{i} \neq i, T_{\hat{i},\hat{k}} \cap T_{i,k} \neq \emptyset\}. \tag{24}$$

Based on Eq. (24), recombination of the task dependency graph of PB won't select Jack's $u^{\text{v}}_{1,1}$ as the critical user. Thus, Jack cannot improve his utility strategically.

(3) Theoretical analysis: we analyze the strategy-proof of the above payment design and have Theorem 4.

Theorem 4. *Picasso is individually rational and truthful, both of which are called strategy-proof.*

Proof. In what follows, we will prove the individual rationality and the truthfulness one by one.

Proof of individual rationality: for each u_i, if $u^{\text{v}}_{i,k}$ is not selected, according to Alg. 1, $p_{i,k} = 0$, $c_{i,k} = 0$. Otherwise, $p_{i,k} = a_{\hat{i}^*,\hat{k}^*} \cdot \sqrt{|T_{i,k}|}/\sqrt{|T_{\hat{i}^*,\hat{k}^*}|}$. As $\sqrt{|T_{i,k}|}/a_{i,k} \geq \sqrt{|T_{\hat{i}^*,\hat{k}^*}|}/a_{\hat{i}^*,\hat{k}^*}$, $p_{i,k} \geq a_{i,k}$. Since $a_{i,k} \geq c_{i,k}$, u_i's utility $\sum_{k=1}^{N_i}(p_{i,k} - c_{i,k}) \geq 0$. Thus, *Picasso* is individually rational.

Proof of truthfulness: we prove it in terms of (i) independent SMBs which are proved truthful in many existing studies [15], [24], [53], and (ii) dependent SMBs. In what follows, we prove that *Picasso* is still truthful even with dependent SMBs, using *reductio ad absurdum* method.

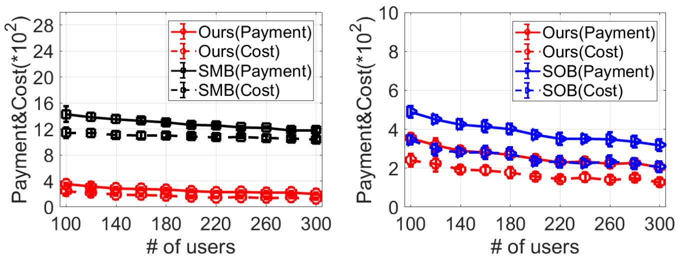
We assume that the original proposition is not true, *i.e.,* there exists a user (say u_i) who can improve his utility by unilaterally misreporting his costs. Specifically, u_i improves his utility by changing the bids of his virtual users $u^{\text{v}}_{i,k}(k = 1 \ldots N_i)$ from $B_i = \{(T_{i,k}, c_{i,k}) | k = 1, \ldots, \delta_i\}$ to $\widetilde{B}_i = \{(T_{i,k}, a_{i,k}) | k = 1, \ldots, \delta_i\}$, where $(c_{i,1}, \ldots, c_{i,\delta_i}) \neq (a_{i,1}, \ldots, a_{i,\delta_i})$. We prove this theorem in terms of two different cases, *i.e.,* u_i is unselected/ selected with B_i. The detailed proofs are provided in the Appendix.

□

In summary, even if users strategically use the task dependencies, *Picasso* achieves the truthfulness and the individual rationality by recombining the task dependency graph of PB.

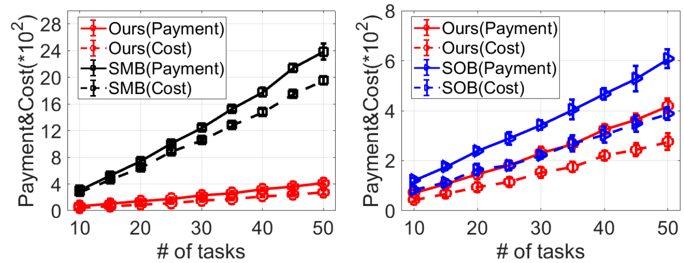
5 PERFORMANCE EVALUATION OF *Picasso*

We first conduct extensive simulations to evaluate the performance of *Picasso*, which is further tested by conducting a real case study of Gigwalk based on real traces.



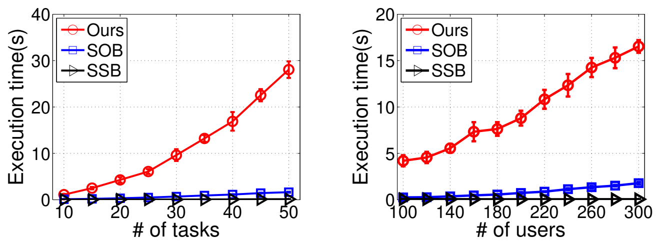
(a) *Picasso* vs SMB (b) *Picasso* vs SOB

Fig. 7: Comparison of the social cost and the total payment for different numbers of users.



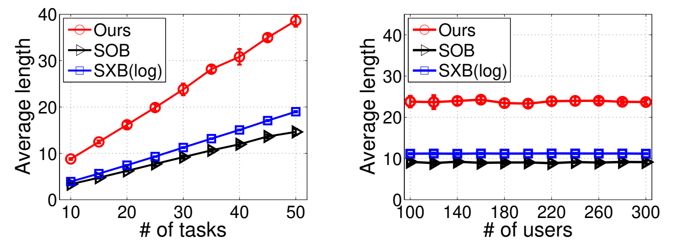
(a) *Picasso* vs SMB (b) *Picasso* vs SOB

Fig. 8: Comparison of the social cost and the total payment for different numbers of tasks.



(a) Different number of tasks (b) Different number of users

Fig. 9: Comparison of time cost with different numbers of users and tasks by comparing SSB and SOB.



(a) Different number of tasks (b) Different number of users

Fig. 10: Comparison of average description length (ADL) for different numbers of users and tasks.

5.1 Simulations

5.1.1 Simulation Methodology & Settings

There are N users to provide PBs for M tasks. The number L of OR bids for one PB and the number K of atomic bids in one OR bid are both uniformly distributed, *i.e.*, $L \sim \mathrm{U}(1, R)$ ($R = 5$) and $K \sim \mathrm{U}(1, 0.6M)$. The real cost of each user in executing a task is normally distributed as $\mathrm{N}(\mu, \sigma^2)$, where $\mu \sim \mathrm{U}(20, 40)$ and $\sigma \sim \mathrm{U}(5, 15)$. Each data point is obtained by averaging 20 execution results. Our simulation has been conducted on a PC with 2.3 GHz dual-core Intel Core i5 CPU and 8 GB RAM. *Picasso* is compared to the aforementioned four baseline methods, **SMB** [24], **SOB** [15], **SXB** [23], [50], and **OPT**. SMB, SOB, and SXB use the greedy algorithm and the critical payment in task allocation and payment computation which are similar to *Picasso*. OPT utilizes brute-force search and Vickrey Clarke Groves mechanism [34] for the optimal solution using the same description method as *Picasso*. We use four performance metrics, *i.e.*, *social cost*, *total payment*, *time cost*, and *ADL*.

5.1.2 Results

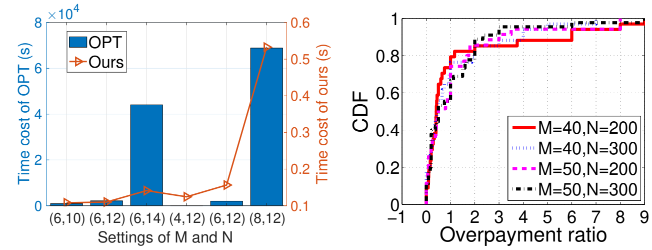
We first compare *Picasso* with SMB and SOB in terms of social cost and total payment for different numbers of users. We set $M = 30$ and vary N from 100 to 300. As shown in Figs. 7a and 7b, the social cost and total payment of *Picasso* are always less than those of SMB and SOB by a large margin. In terms of both social cost and total payment, *Picasso* outperforms SMB and SOB by more than 32.6% for a varying number of users. We also vary M from 10 to 50 and set $N = 200$. Figs. 8a and 8b show that the social cost and total payment of *Picasso* are always much smaller than those of SMB and SOB. *Picasso* outperforms SMB and SOB in both social cost and total payment by more than 34.9% for different numbers of tasks.

We evaluate the computation time for different numbers of users. Figs. 9b and 9a show that although the time cost of *Picasso* is higher than that of SMB and SOB, it increases roughly quadratically with N and M. In particular, *Picasso* costs only 16.5 s in the worst case and 9.8 s on average when the number of users is changed from 100 to 300. Also, it costs 28.1 s in the worst case and 11.6 s on average when the number of tasks is changed from 10 to 50. We also compare the execution times of *Picasso* and OPT in Fig. 11a. As the problem is NP-hard, OPT takes significantly long (the time complexity is $\mathcal{O}(M^M N^M)$), *e.g.*, running for more than 68,859 s (about 19 h) only when $M = 8$ and $N = 12$, while *Picasso* completes within 0.6 s, which can be negligible in practice. Moreover, its execution time sharply increases with the number of tasks and users, making it much less applicable to large-scale systems. These results are consistent with the theoretical analysis in Lemma 2.

We use the ADL in Figs. 10a and 10b to evaluate the description efficiency of *Picasso* in comparison with SOB and SXB. As SMB has much worse expressiveness than SOB and its ADL is always 1, we do not include it here. As the ADL of SXB is extremely large, we show its logarithm for ease of presentation. Fig. 10a shows that ADL increases linearly with the number of tasks for both *Picasso* and SOB, while it increases exponentially with the number of tasks for SXB. ADL of SOB and *Picasso* are 9.1 and 23.7 on average, respectively, while that of SXB rises dramatically up to $2^{16.9}$. As illustrated in Fig. 10b, we also observe that ADL changes slightly with the number of users for all of these three methods. ADL of SOB and *Picasso* are 9.1 and 23.8 on average, respectively, while that of SXB explodes to $2^{11.2}$. These results are consistent with the theoretical analysis in Sec. 4.1.2. In conclusion, *Picasso* is more description-efficient than SXB, achieving an excellent trade-off among expressiveness, description efficiency, and computational complexity.

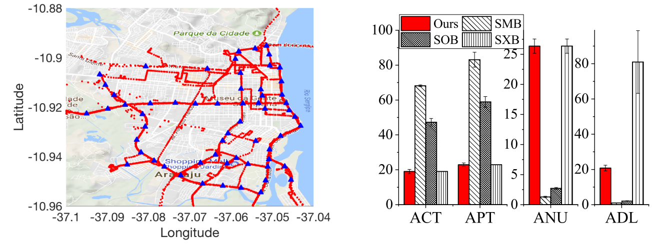
Finally, we evaluate the individual rationality of our method. We plot the CDF for the ratio of user's extra payment to her/his real cost called *overpayment ratio* for four

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/JIOT.2020.3044731, IEEE Internet of Things Journal



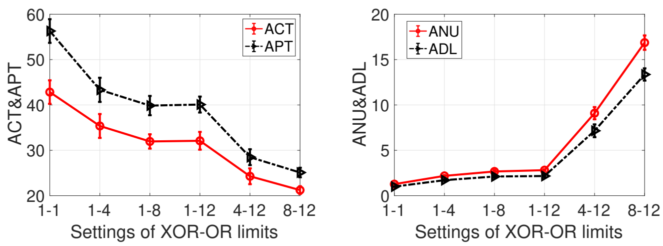
(a) Time cost (b) Overpayment ratio

Fig. 11: (a) Comparison of time costs of *Picasso* and OPT. (b) CDF of overpayment ratio.



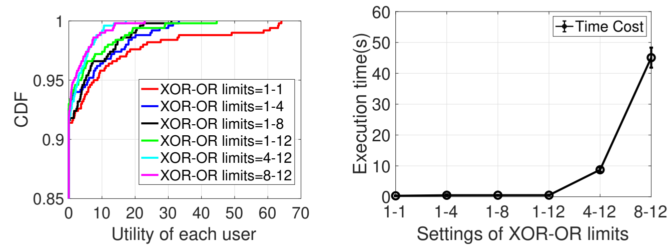
(a) Trajectories of 200 drivers (b) Performance evaluation

Fig. 12: Trace-based Gigwalk case studies: a) Driver's trajectories; b) Performance comparisons.



(a) ACT and APT (b) ANU and ADL

Fig. 13: Performance evaluation for different settings of XOR and OR limits, including ACT, APT, ANU, and ADL.



(a) Users' utility (b) Time cost

Fig. 14: Evaluation of users' utility and system's time cost with different settings of the XOR-OR limits.

different settings (*i.e.*, user and task number). As illustrated in Fig. 11b, the overpayment ratio is above 0 for 100%, below 1 for 60%, and below 2 for 80%. The results show that all of the payments for users are more than their real cost. Moreover, the overpayment, *i.e.*, within double of real cost for 80%, is reasonable. Thus, our method achieves individual rationality.

5.2 Trace-Driven Case Study of Gigwalk

5.2.1 Evaluation Methodology & Settings

To strengthen our evaluation, we have conducted a case study, emulating Gigwalk based on the real traces of drivers[3] with the following setup.

Traces & Participants: as shown in Fig. 12a, we use the real trajectories of 200 drivers to emulate the mobility of users, and randomly select 20 locations (represented by the blue triangles) in these trajectories. Each location has a random number of tasks, following the distribution $\mathrm{U}(1, 5)$. For example, there are multiple different tasks at a shop or in its neighborhoods. We emulate 500 crowdsourcing participants expecting the Gigwalk tasks on their driving paths. Each participant randomly chooses a starting point, such as her/his home or work location. We use the real driving paths within 3 km from the participant's starting point as her/his potential paths, along only one of which s/he will drive.

Time & Price: it takes time for each participant to do the tasks, as one needs to find a parking lot [1] and then picks up the task. We set 10 min per location on average as her/his dwelling time according to the report in [1]. Moreover, different users have different available time limits following $\mathcal{U}(10, 120)$ (min). Each user randomly chooses a limited number of locations up to her/his time limit on her/his path, and performs the tasks at any subset of these locations. At each location, a participant chooses a random number of tasks, and expects to do either all of them after parking there, or none without stopping, owing to her/his own rationality. The bidding price for each task is initiated according to the user, driving paths, and locations within a distribution $\mathcal{U}(\$1, \$100)$, and its actual price depends on the task type, platform, and area. For example, if the platform sets the maximum price \$10, the bidding price randomly changes from \$0.10 to \$10.

Metrics: besides **ADL** for description efficiency, we use the **A**verage bidding task **N**umber per **U**ser (**ANU**) to evaluate the user's expressiveness. The more tasks each user can bid, the more user's expressiveness this mechanism enables. On the other hand, we exploit the **A**verage social **C**ost per executed **T**ask (**ACT**) and **A**verage platform **P**ayment per executed **T**ask (**APT**) to represent platform utility.

5.2.2 Results

First, we evaluate the influences made by the settings of XOR-OR limits on the performance of *Picasso*. As the maximum available time of users is set to 120 min and the dwelling time is 10 min, the number of OR operations is no more than 12. Thus, we change the settings of XOR-OR limits $(R_{\mathrm{XOR}}, R_{\mathrm{OR}})$ as (1,1), (1,4), (1,8), (1,12), (4,12) and (8,12), where R_{XOR} and R_{OR} denote the limits of XOR and OR operations, respectively. As illustrated in Figs. 13a and 13b, the ACT and APT of the platform decrease gradually with the limits of XOR and OR operations, while both ANU and ADL are increasing with them. On the other hand, as shown in Figs. 14a and 14b, the high XOR-OR limits diminish the utility of each user, and increase the execution time of the proposed algorithm. The results indicate that the higher XOR-OR limits bring more bidding freedom, encouraging users to bid more tasks, which increase the description length and the computation time. However, both the ADL and the time cost of *Picasso* grow slowly with the XOR-

3. The traces are collected by a smartphone app called Go!Track, including 200 trajectories with 16,664 GPS coordinates from 200 different drivers. https://archive.ics.uci.edu/ml/datasets/GPS+Trajectories.

OR limits, which is a polynomial increase as illustrated in Figs. 13b and 14b. Interestingly, more bidding freedom stimulates competition between users for the allocation of limited tasks, decreasing the utility of each user, hence resulting in much lower social cost (and platform payment). Similar to economic freedom, it empowers people and unleashes their powerful forces of choice, thus enhancing the market competition and improving the overall economy [4].

Moreover, we compare *Picasso* with existing schemes, *i.e.*, SMB, SOB, and SXB. The presented means are out of 20 emulations. Fig. 12b shows that, compared to SMB and SOB, *Picasso* reduces ACT and APT by more than 60% and 61%, respectively, while increasing ANU by at least 9.7x. So, *Picasso* effectively enables the user's expressiveness and significantly reduces the social cost as well as the platform payment. On the other hand, compared to SXB, *Picasso* cuts ADL by more than 74%, despite their similar ACT, APT, and ANU. Thus, *Picasso* achieves powerful expressiveness without compromising description efficiency.

In summary, *Picasso* is shown to benefit not only the platform owner by significantly lowering its payment, but also the participants by raising their intrinsic motivation with more expressiveness and description efficiency. That is, *these Gigwalk case studies confirmed the effectiveness of Picasso in incentivizing both the platform owner and the participants*.

6 Discussion & Future Work

We discuss the influences of practical factors in crowdsensing applications on *Picasso* as follows.

Task execution unreliability: After task bidding and allocation, the users may fail to execute these allocated tasks, owing to users' unreliability and their mobility's uncertainty [16]. To address this issue of task execution unreliability, *Picasso* only rewards the users who successfully execute the tasks, such as uploading the sensed data. Furthermore, to avoid the user's bidding misbehavior (*i.e.*, bidding as many tasks as possible), we can introduce a penalty to the users who do not finish their allocated tasks, such as not allocating tasks to them in the future. In addition, we can extract the users' reliability from their historical behaviors using deep learning methods [14], which is fed back to design the task allocation scheme. In the future, we would like to explore the allocation of tasks based on the users' task execution probability model.

Non-additive cost of tasks: owing to the law of diminishing return in economics [10], a user may pay less cost when conducting multiple tasks together, than the summation of costs when conducting each task individually. For example, in Gigwalk application, when multiple tasks are located very close to each other, some users with enough time may pay a discounted cost when executing all the tasks. However, Picasso can be easily extended to the case of non-additive cost with a simple modification. Specifically, this case happens only when the user wants to finish multiple tasks together, which are represented by the atomic bid in Picasso. Hence, the users can set the minimal cost and the desired payment for this atomic bid according to their actual cost, thus taking account into a diminishing cost. In addition, the diminishing-cost property of the tasks' cost can be further exploited to improve the total utility of the platform owner [49], which is part of our future work.

User-friendly preference expression: In practice, based on the formal bid description in Sec. 4.1, a user-friendly preference expression system should be designed while considering the underlying applications. In what follows, we take the practical personalized bidding in Gigwalk as an example. First, the users input their source and destination locations as well as their available time slots. Then, similarly to the Google navigation, the system returns multiple candidate routes from the source to the destination. Each route consists of several Gigwalk tasks. Moreover, the users express their preferences using user-friendly interfaces, such as binding those tasks expected to be done together. Furthermore, we can utilize machine learning [45] to automatically extract the users' preferences based on their behavior datasets [8], further facilitating the users' input. Finally, the system automatically creates the formal bid description using *Picasso*.

7 Conclusion

We have designed and evaluated a novel PB-based incentive mechanism, called *Picasso*, that consists of two main components. First, we have proposed a PB description method in 3-D expressive space with AND, XOR, and OR, achieving a good trade-off among expressiveness, computational complexity, and description efficiency. Second, we have designed schemes for constant-factor approximation in optimal task allocation and strategy-proof in payment with computational efficiency, by decomposing and recombining task dependency graph of PB. Both the theoretical analysis and trace-based Gigwalk case studies have validated the above essential properties of *Picasso*.

Acknowledgments

This research is supported by the NSF of China Projects: Grants No.61872447, No.62072303, the Natural Science Foundation of Chongqing (No. CSTC2018JCYJA1879). The corresponding authors are Chaocan Xiang and Panlong Yang.

References

[1] Stop wasting time searching for parking. https://blog.spothero.com/park-smarter-parking-search-time/.

[2] Personalization pulse check 2016. *Accenture Interactive(Report)*, 2016.

[3] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach.* Cambridge University Press, 2009.

[4] Niclas Berggren. The Benefits of Economic Freedom: A Survey. *The Independent Review*, 8:193–211, 2003.

[5] Craig Boutilier. Solving Concisely Expressed Combinatorial Auction Problems. In *Pro. AAAI*, 2002.

[6] Craig Boutilier and Holger H Hoos. Bidding Languages for Combinatorial Auctions. In *Proc. IJCAI*, 2001.

[7] Zhipeng Cai, Zhuojun Duan, and Wei Li. Exploiting Multi-Dimensional Task Diversity in Distributed Auctions for Mobile Crowdsensing. *IEEE Transactions on Mobile Computing*, 2020.

[8] Chao Chen, Daqing Zhang, Xiaojuan Ma, Bin Guo, Leye Wang, Yasha Wang, and Edwin Sha. Crowddeliver: Planning city-wide package delivery paths leveraging the crowd of taxis. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1478–1496, 2016.

[9] Yanjiao Chen, Baochun Li, and Qian Zhang. Incentivizing crowdsourcing systems with network effects. In *Proc. IEEE INFOCOM*, pages 1–9, 2016.

[10] Hsin-Hung Chou, Hsuan-Chao Chiu, Nigel F. Delaney, Daniel Segre, and Christopher J. Marx. Diminishing returns epistasis among beneficial mutations decelerates adaptation. *Science*, 281:99–102, 1998.

[11] Lingjie Duan, Longbo Huang, Cedric Langbort, Alexey Pozdnukhov, Jean Walrand, and Lin Zhang. Human-in-the-loop mobile networks: A survey of recent advancements. *IEEE J-SAC*, 35(4):813–831, 2017.

[12] Zhuojun Duan, Wei Li, and Zhipeng Cai. Distributed Auctions for Task Assignment and Scheduling in Mobile Crowdsensing Systems. In *Proc. IEEE ICDCS*, pages 635–644, 2017.

[13] Zhuojun Duan, Wei Li, Xu Zheng, and Zhipeng Cai. Mutual-preference driven truthful auction mechanism in mobile crowdsensing. In *Proc. IEEE ICDCS*, pages 1233–1242, 2019.

[14] Xiaochen Fan, Chaocan Xiang, Chao Chen, Xudong Song, Panlong Yang, Liangyi Gong, Priyadarsi Nanda, and Xiangjiang He. Buildsensys: Reusing building sensing data for traffic prediction with cross-domain learning. *IEEE Transactions on Mobile Computing*, 2020.

[15] Zhenni Feng, Yanmin Zhu, Qian Zhang, Lionel M Ni, and Athanasios V Vasilakos. Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *Proc. IEEE INFOCOM*, pages 1231–1239, 2014.

[16] Guoju Gao, Mingjun Xiao, Jie Wu, Liusheng Huang, and Chang Hu. Truthful Incentive Mechanism for Nondeterministic Crowdsensing with Vehicles. *IEEE Transactions on Mobile Computing*, 17(12):2982–2997, 2018.

[17] David Geiger and Martin Schader. Personalized task recommendation in crowdsourcing information systems ? current state of the art. *Decision Support Systems*, 65:3 – 16, 2014.

[18] Ronald L Graham. *Handbook of Combinatorics*, volume 1. Elsevier, 1995.

[19] Kai Han, Yuntian He, Haisheng Tan, Shaojie Tang, He Huang, and Jun Luo. Online Pricing for Mobile Crowdsourcing with Multi-Minded Users. In *Proc. ACM MobiHoc*, 2017.

[20] Kai Han, He Huang, and Jun Luo. Quality-Aware Pricing for Mobile Crowdsensing. *IEEE TON*, 26(4):1728–1741, 2018.

[21] Suining He and Kang G Shin. (re) configuring bike station network via crowdsourced information fusion and joint optimization. In *Proc. ACM Mobihoc*, pages 1–10, 2018.

[22] Suining He and Kang G Shin. Steering crowdsourced signal map construction via bayesian compressive sensing. In *Proc. IEEE INFOCOM*, pages 1016–1024, 2018.

[23] Haiming Jin, Lu Su, Danyang Chen, Hongpeng Guo, Klara Nahrstedt, and Jinhui Xu. Thanos: Incentive Mechanism with Quality Awareness for Mobile Crowd Sensing. *IEEE TMC*, 18(8):1951–1964, 2019.

[24] Haiming Jin, Lu Su, and Klara Nahrstedt. CENTURION: Incentivizing multi-requester mobile crowd sensing. In *Proc. IEEE INFOCOM*, 2017.

[25] Haiming Jin, Lu Su, Houping Xiao, and Klara Nahrstedt. Incentive Mechanism for Privacy-Aware Data Aggregation in Mobile Crowd Sensing Systems. *IEEE TON*, 26(5):2019–2032, 2018.

[26] Xiaocong Jin and Yanchao Zhang. Privacy-Preserving Crowdsourced Spectrum Sensing. *IEEE TON*, 26(3):1236–1249, 2018.

[27] Iordanis Koutsopoulos. Optimal incentive-driven design of participatory sensing systems. In *Proc. IEEE INFOCOM*, pages 1402–1410, 2013.

[28] Sébastien Lahaie, David C. Parkes, and David M. Pennock. An Expressive Auction Design for Online Display Advertising. In *Pro. AAAI*, pages 108–113, 2008.

[29] Ji Li, Zhipeng Cai, Mingyuan Yan, and Yingshu Li. Using crowdsourced data in location-based social networks to explore influence maximization. In *Proc. IEEE INFOCOM*, 2016.

[30] Jian Lin, Ming Li, Dejun Yang, Guoliang Xue, and Jian Tang. Sybil-Proof Incentive Mechanisms for Crowdsensing. In *Proc. IEEE INFOCOM*, pages 1–9, 2017.

[31] Yaguang Lin, Zhipeng Cai, Xiaoming Wang, and Fei Hao. Incentive Mechanisms for Crowdblocking Rumors in Mobile Social Networks. *IEEE Transactions on Vehicular Technology*, 68(9):9220–9232, 2019.

[32] Tie Luo, Hwee-Pink Tan, and Lirong Xia. Profit-maximizing incentive for participatory sensing. In *Proc. IEEE INFOCOM*, pages 127–135, 2014.

[33] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proc. ACM EC*, pages 1–12, 2000.

[34] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic Game Theory*, volume 1. Cambridge University Press, 2007.

[35] Yuben Qu, Shaojie Tang, Chao Dong, Peng Li, Song Guo, Haipeng Dai, and Fan Wu. Posted Pricing for Chance Constrained Robust Crowdsensing. *IEEE TMC*, pages 1–1, 2019.

[36] Francesco Restuccia, Sajal K Das, and Jamie Payton. Incentive mechanisms for participatory sensing: Survey and research challenges. *ACM TOSN*, 12(2):13, 2016.

[37] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1):54–67, 2000.

[38] Ming Tang, Haitian Pang, Shou Wang, Lin Gao, Jianwei Huang, and Lifeng Sun. Multi-Dimensional Auction Mechanisms for Crowdsourced Mobile Video Streaming. *IEEE TON*, 26:2062–2075, 2018.

[39] Jiangtao Wang, Feng Wang, Yasha Wang, Leye Wang, Zhaopeng Qiu, Daqing Zhang, Bin Guo, and Qin Lv. HyTasker: Hybrid Task Allocation in Mobile Crowd Sensing. *IEEE Transactions on Mobile Computing*, 19(3):598–611, 2020.

[40] Jiangtao Wang, Feng Wang, Yasha Wang, Daqing Zhang, Leye Wang, and Zhaopeng Qiu. Social-Network-Assisted Worker Recruitment in Mobile Crowd Sensing. *IEEE Transactions on Mobile Computing*, 18(7):1661–1673, 2019.

[41] Jiangtao Wang, Yasha Wang, Daqing Zhang, Feng Wang, Haoyi Xiong, Chao Chen, Qin Lv, and Zhaopeng Qiu. Multi-Task Allocation in Mobile Crowd Sensing with Individual Task Quality Assurance. *IEEE TMC*, 17(9):2101–2113, 2018.

[42] Zhibo Wang, Jiahui Hu, Ruizhao Lv, Jian Wei, Qian Wang, Dejun Yang, and Hairong Qi. Personalized Privacy-Preserving Task Allocation for Mobile Crowdsensing. *IEEE TMC*, 18(6):1330–1341, 2019.

[43] Chaocan Xiang, Panlong Yang, Chang Tian, Haibin Cai, and Yunhao Liu. Calibrate without calibrating: An iterative approach in participatory sensing network. *IEEE Transactions on Parallel and Distributed Systems*, 26(2):351–361, 2015.

[44] Chaocan Xiang, Panlong Yang, Chang Tian, Lan Zhang, Hao Lin, Fu Xiao, Maotian Zhang, and Yunhao Liu. Carm: crowd-sensing accurate outdoor rss maps with error-prone smartphone measurements. *IEEE Transactions on Mobile Computing*, 15(11):2669–2681, 2016.

[45] Chaocan Xiang, Zhao Zhang, Yuben Qu, Dongyu Lu, Xiaochen Fan, Panglong Yang, and Fan Wu. Edge Computing-Empowered Large-scale Traffic Data Recovery Leveraging Low-rank Theory. *IEEE Transactions on Network Science and Engineering*, pages 1–1, 2020.

[46] Jia Xu, Chengcheng Guan, Haobo Wu, Dejun Yang, Lijie Xu, and Tao Li. Online incentive mechanism for mobile crowdsourcing based on two-tiered social crowdsourcing architecture. In *Proc. IEEE SECON*, pages 1–9, 2018.

[47] Jia Xu, Zhengqiang Rao, Lijie Xu, Dejun Yang, and Tao Li. Incentive Mechanism for Multiple Cooperative Tasks with Compatible Users in Mobile Crowd Sensing via Online Communities. *IEEE Transactions on Mobile Computing*, 19(7):1618–1633, 2019.

[48] Jinliang Xu, Shangguang Wang, Ning Zhang, Fangchun Yang, and Xuemin Shen. Reward or Penalty: Aligning Incentives of Stakeholders in Crowdsourcing. *IEEE TMC*, 18(4):974–985, 2019.

[49] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proc. ACM MobiCom*, 2012.

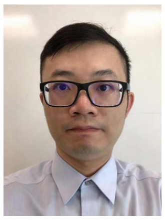
[50] Xiang Zhang, Guoliang Xue, Ruozhou Yu, Dejun Yang, and Jian Tang. Truthful incentive mechanisms for crowdsourcing. In *Proc. IEEE INFOCOM*, pages 2830–2838, 2015.

[51] Xinglin Zhang, Zheng Yang, Wei Sun, Yunhao Liu, Shaohua Tang, Kai Xing, and Xufei Mao. Incentives for mobile crowd sensing: A survey. *IEEE COMST*, 18(1):54–67, 2016.

[52] Dong Zhao, Xiang-Yang Li, and Huadong Ma. Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully. *IEEE ToN*, 24(2):647–661, 2016.

[53] Zhenzhe Zheng, Fan Wu, Xiaofeng Gao, Hongzi Zhu, Guihai Chen, and Shaojie Tang. A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing. *IEEE TMC*, 2017.

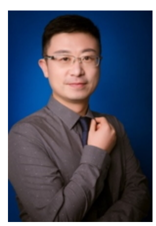
Chaocan Xiang is currently an associate professor at College of Computer Science, Chongqing University, Chongqing, China. He received the BS and PhD degrees in computer science and engineering from the Nanjing Institute of Communication Engineering, China, in 2009 and 2014, respectively. He studied in University of Michigan-Ann Arbor in 2017. His current research interests include wireless sensor networks, crowd-sensing networks, and IoT.



Suining He is currently working as an assistant professor at the Department of Computer Science and Engineering, the University of Connecticut. He received his PhD degree in Computer Science and Engineering at the Hong Kong University of Science and Technology, and worked as a postdoctoral research fellow at the Real-Time Computing Lab (RTCL), the University of Michigan, Ann Arbor. His research interests include ubiquitous and mobile computing, crowdsourcing and big data analytics.

Kang G. Shin (LF'12) is currently the Kevin and Nancy O'Connor Professor of computer science and the Founding Director of the Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. He received the B.S. degree in electronics engineering from Seoul National University, Seoul, South Korea, and the M.S. and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, USA, in 1970, 1976, and 1978, respectively. At the University of Michigan, he has supervised the completion of 82 Ph.D. students and also chaired the Computer Science and Engineering Division for three years starting in 1991. From 1978 to 1982, he was with the faculty of the Rensselaer Polytechnic Institute, Troy, NY, USA. He has authored/coauthored more than 900 technical articles (more than 330 of which are published in archival journals) and holds more than 30 patents or invention disclosures. His current research interests focus on QoS-sensitive computing and networks as well as on embedded real- time and cyber-physical systems. He has also received numerous institutional awards and the best paper awards. He is a fellow of ACM.

Yuben Qu received the B.S. degree in Mathematics and Applied Mathematics from Nanjing University, and both the M.S. degree in Communication and Information Systems and the Ph.D degree in Computer Science and Technology from Nanjing Institute of Communication Engineering, China, in 2009, 2012 and 2016, respectively. He is currently a postdoc in Shanghai Jiao Tong University, China. From October 2015 to January 2016, he was a visiting research associate in the School of Computer Science and Engineering, the University of Aizu, Japan. His research interests include D2D communications, edge computing, crowdsensing and network coding.



Panlong Yang is now a professor in the School of Computer Science and Technology, University of Science and Technology of China. He received the BS, MS, and Ph.D. degrees in communication and information system from the Nanjing Institute of Communication Engineering, China, in 1999, 2002, and 2005, respectively. His research interests include Mobile/Pervasive Computing, IoT.