Noname manuscript No.
(will be inserted by the editor)

A Budget Feasible Peer Graded Mechanism For IoT-Based Crowdsourcing

Vikash Kumar Singh · Sajal Mukhopadhyay · Fatos Xhafa · Aniruddh Sharma

Received: date / Accepted: date

Abstract We develop and extend a line of recent works on the design of mechanisms for heterogeneous tasks assignment problem in 'crowdsourcing'. The budgeted market we consider consists of multiple task requesters and multiple IoT devices as task executers. In this, each task requester is endowed with a single distinct task along with the publicly known budget. Also, each IoT device has valuations as the cost for executing the tasks and quality, which are private. Given such scenario, the objective is to select a subset of IoT devices for each task, such that the total payment made is within the allotted quota of the budget while attaining a threshold quality. For the purpose of determining the unknown quality of the IoT devices we have utilized the concept of *peer grading*. In this paper, we have carefully crafted a *truthful budget feasible* mechanism for the problem under investigation that also allows us to have the true information about the *quality* of the IoT devices. Further, we have extended the set-up considering the case where the tasks are divisible in nature and the IoT devices are working collaboratively, instead of, a single entity for executing each task. We have designed the *budget feasible* mechanisms for

V. K. Singh
Department of Computer Science & Engineering, National Institute of Technology Durgapur, West Bengal, India
E-mail: vikas.1688@gmail.com

S. Mukhopadhyay
Department of Computer Science & Engineering, National Institute of Technology Durgapur, West Bengal, India

F. Xhafa
Department of Computer Science, Universitat Politècnica de Catalunya, Barcelona, Spain
E-mail: fatos@cs.upc.edu

A. Sharma
Engineer, Qualcomm India Private Ltd., Hyderabad, Andhra Pradesh, India

the extended versions. The simulations are performed in order to measure the efficacy of our proposed mechanism.

Keywords Crowdsourcing · IoT devices · Truthful · Budget feasible · Peer grading · Shapley value

1 Introduction

Over the past decades, most of the works in *crowdsourcing* Howe (2006); Slivkins and Vaughan (2014) mainly circumvent around tackling one of the major challenges of *how to motivate the crowd workers to participate in the system?* One solution that has been appreciated a lot in this direction is, to incentivize the task executers. This gave rise to several other open questions: 1) Which task executers to be hired? 2) How the task requester(s) can be aware about the quality of the task executers (or crowd workers)? 3) What amount is to be paid to the task executers for their services, so that they are not dishearten and are motivated to participate in future in similar type of systems? Answering to the above raised questions, substantial amount of works have been done in these directions Bhat et al. (2016); Gao et al. (2015b); Goel et al. (2014); Jain et al. (2018, 2016); Luo et al. (2016); Chatzimilioudis et al. (2012). In this paper, we have investigated the set-up motivated by the set-ups discussed in Goel et al. (2014); Assadi et al. (2015). In our setup: 1) the task executers are the IoT devices instead of human agents, and 2) in order to be aware about the quality of IoT devices, we have utilized the technique of peer grading. It is different from the general practice for identifying the quality of the human agents Jain et al. (2018); Bhat et al. (2016). It is to be noted that, till date, in the crowdsourcing literature this tedious job of determining the quality of the crowd workers is mostly done by the platform or in some cases by the task requesters. This leads to an extra burden on the platform or the task requesters. Also, this scenario makes the process of quality determination centralized. In our peer grading approach, we use to distribute the task executed by the IoT devices to their peers (other IoT devices), for grading purpose. Based on the peers report, the quality IoT devices are selected.

The detailing of our proposed model is depicted in Fig. 1. In our model, we have multiple task requesters and multiple IoT devices (as task executers). Each task requester is endowed with a single task and the maximum amount he/she (henceforth he) can pay termed as *budget* (or capital). Each IoT device has an independent private cost for each task, that they will charge for executing the task. It is to be noted that, the participating IoT devices are intelligent and rational. Due to their rational behaviour they will try to strategize the system. By *strategizing* we mean that these devices can manipulate their private information(s) in order to gain. Given this set-up, our goal is to select the subset of IoT devices for each task such that the total payment made to the IoT devices is within the allotted quota of budget for the task while attaining a threshold quality. Following the general work flow of the crowdsourcing, firstly, each task requester submits the endowed task and the publicly known budget to the platform. On receiving the tasks and the endowed capital for the respective task from the task requesters, the platform publishes the tasks to the outside world for the execution purpose.

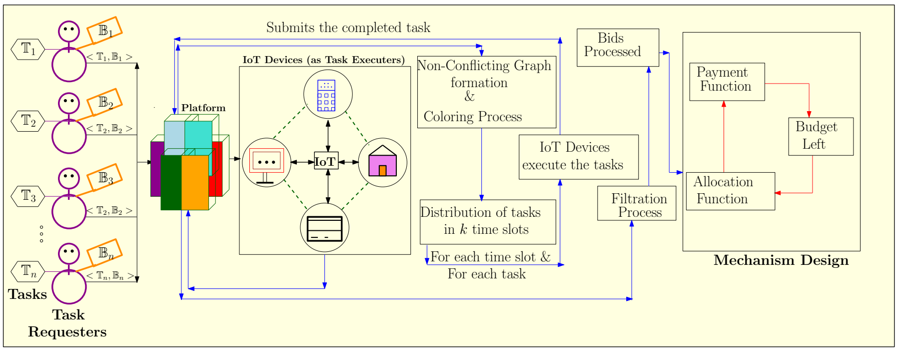


Fig. 1: Pictorial representation of proposed model

Now, each IoT device presents on the other side of the market, opts for the subset of tasks of their interest for execution. They report their interest to the platform along with the amount they will charge for executing each task. Based on their reported interests, the platform assigns the tasks to the IoT devices. In our set-up, it is assumed that each IoT device will execute all of its tasks for which it has shown interest and each IoT device executes single task at a time. Now, the immediate question is: *How to preserve the assumptions made for the problem under investigation?* One solution that could be thought of, is to place each of the tasks of an IoT device on which it has shown interest into different time slots (here, time slots could be thought of as *morning*, *afternoon*, and *evening* for a day) that will help in keeping our assumptions alive.

Say, for example an IoT device has shown its interest over 3 tasks. In such case, one task will be scheduled in the morning, another task in the afternoon, and the last one will be scheduled in the evening.

After the distribution of tasks into different time slots, the IoT devices executes the assigned task(s) and submit to the platform as depicted in Fig. 1. Now, the next challenge that comes into the pocket of the *platform* is to determine the quality of the IoT devices. For this purpose, the idea of peer grading Alfaro et al. (2016); Roughgarden (2016) is utilized in our set-up. It is to be noted that, in each time slot and for each task, the process of *peer grading* is carried out. The process continues until each IoT device is not graded by its peers. Finally, the peer grading process returns a set of quality IoT devices for each task. Now, given the set of quality IoT devices for each task, we have to select a subset of IoT devices in a way that the total payment made is within the allotted quota of budget.

In this paper, we have carefully crafted a *truthful budget feasible mechanism* for the *task allocation problem* (TUBE-TAP) motivated by Singer (2010); Singh et al. (2018b), that also allow us to have the true information about the *quality* of the IoT devices[1]. Further in this line, we have extended our rudimentary model by injecting the constraint that the tasks endowed by the task requesters are divisible in nature. It is to be noted that, the solution approach for the later version of the models differs from the rudimentary model only in terms of mechanism design part. In this context, *non-truthful budget feasible mechanisms* are proposed motivated by Maschler et al. (2013); Shapley (1953); Singer (2010) to cater the need of the problems to some extent.

The main contributions of this paper are:

- We have investigated the heterogeneous task assignment problem in IoT based crowdsourcing through the lens of mechanism design.
- We have developed a *truthful budget feasible* mechanism and the *non-truthful budget feasible* mechanisms for the rudimentary version and the more realistic versions of the problem respectively.
- We prove that TUBE-TAP is *truthful* and *budget feasible* through simulation and theoretical analysis.
- The simulations are done for comparing the TUBE-TAP with a carefully crafted benchmark mechanism.

The remainder of this paper is organized as follows. In section 2 the prior works explored in the direction of crowdsourcing are discussed. In section 3, we describe our proposed system model in detailed manner. We then present our proposed mechanism namely TUBE-TAP in section 4. Further analysis of TUBE-TAP is carried out in section 5. In section 6 the more general setting with divisible task is discussed in detailed manner. Further enhancement of the model is done in section 7. In section 8 the experimental results are presented and discussed. In section 9 the paper is concluded and the future directions are coined.

2 Related Works

In order to get the detailed overview of the crowdsourcing we recommend readers to go through Howe (2006); Yuen et al. (2011); Slivkins and Vaughan (2014); Mazlan et al. (2018); Daniel et al. (2018). In past there have been an extensive body of works discussing about the major challenges in crowdsourcing Slivkins and Vaughan (2014) and in some cases providing the solution approach Bhat et al. (2016); Jain et al. (2016); Luo et al. (2016). The two major challenges in crowdsourcing that have dragged the interest of large community are: 1) How to motivate large group of common people to participate in this system, as they are *rational*. 2) How to verify that the executed tasks supplied

[1] It is to be noted, our proposed system is applicable equally to the system where there are human agents instead of IoT devices in the role of *task executers*.

by the agents are upto the mark. Answering to the issue raised in point 1 several schemes are proposed that incentivize the participating agents in some sense Luo et al. (2016); Goel et al. (2014); Duan et al. (2017); Li et al. (2018); Reddy et al. (2010); Lee and Hoh (2010b,a). Following works in Reddy et al. (2010), a better auction models were proposed in Zhao et al. (2014); Gao et al. (2015a); Feng et al. (2014). In Goel et al. (2014), an effort has been made to design a truthful budget feasible mechanism for the set-up consisting of single task requester endowed with multiple tasks and multiple task executers, in an online environment. The task executers along with the private cost have different skills based on which they show their interest to perform certain subset of tasks. The goal is to select the subset of task executers in such a way that the total payment made to the task executers does not exceed the budget. In the similar line, the work by Xu et al. (2017) is carried out where, the set-up consists of multiple tasks with deadlines that are to be executed by the pool of workers that arrive online. Each of the workers has the preferred set of tasks that he/she can perform and based on that the task is assigned to the workers before its deadline. The goal is to design an online-assignment policy such that the total expected profit is maximized subject to budget and deadline constraints. In Tinati et al. (2017) discussion regarding several open research questions associated with IoT is made. Also, the emphasis is made on how the crowdsourcing and the IoT could be meld together to resolve several challenging aspects associated with the IoT.

However, the literature covered till now, in this paper, does not consider the quality of the data supplied or more formally, the quality of the crowd workers. Some quality adaptive schemes are discussed in Jain et al. (2018); Gao et al. (2015b); Gong and Shroff (2018). In Kobayashi et al. (2018) the two stage scheme is used for improving the quality of the crowd workers termed as *self-correction*. In the first stage, the workers execute the supplied tasks and submit the executed tasks. In the second stage, the workers review the executed task by other workers, may update their result accordingly, and resubmit the improved version of the executed task.

3 System Model and Problem Formulation

In this section, we present the formal statement of our problem. We consider n task requesters $\mathbb{R} = \{\mathbb{R}_1, \mathbb{R}_2, \ldots, \mathbb{R}_n\}$ each carrying a single distinct task. The set of tasks is represented as $\mathbb{T} = \{\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_n\}$; where $\mathbb{T}_i$ is the i^{th} task held by $\mathbb{R}_i$ task requester. Also, along with a task, each task requester $\mathbb{R}_i \in \mathbb{R}$ has an upper bound on the amount he/she (henceforth he) can pay for getting his task executed, known as *budget* represented as $\mathbb{B}_i$. The budget vector for all the task requesters is given as $\mathbb{B} = \{\mathbb{B}_1, \mathbb{B}_2, \ldots, \mathbb{B}_n\}$. Each of the task requester submits the endowed task along with their publicly known budget to the *platform*. The *platform* projects these tasks to the IoT devices present on the other side of the market. In our set-up, we have m IoT devices represented by the set $\mathbb{E} = \{\mathbb{E}_1, \mathbb{E}_2, \ldots, \mathbb{E}_m\}$. It is considered that $m \gg n$. Afterwards, each IoT device shows its interest over the set of tasks for execution purpose to the platform along with the maximum value it can charge for executing each task. Utilizing the submitted information by the IoT devices, we can have the set of IoT devices that are interested to execute the task $\mathbb{T}_j$ and is given as $\boldsymbol{I}^j = \{\mathbb{E}_1, \mathbb{E}_2, \ldots, \mathbb{E}_{k_j}\}$; where k_j is the number of IoT devices associated with task $\mathbb{T}_j$. The set $\boldsymbol{I} = \{\boldsymbol{I}^1, \boldsymbol{I}^2, \ldots, \boldsymbol{I}^n\}$ represents the associated set of IoT devices for all the n tasks. The maximum value an IoT device $\mathbb{E}_i$ will charge for executing a task $\mathbb{T}_j$ is given as v_i^j called the valuation. The valuations of the IoT devices are *private* in nature. It is to be noted that the IoT devices are *strategic* in nature. By *strategic*, we mean that the IoT devices can misreport their private valuation in order to gain. So, it is better to represent the bid value of each IoT device $\mathbb{E}_i$ for executing the task $\mathbb{T}_j$ as b_i^j. $b_i^j = v_i^j$ represents the fact that the IoT device $\mathbb{E}_i$ report its private valuation b_i^j for the task $\mathbb{T}_j$ in a *truthful* manner. The bid vector for each task $\mathbb{T}_j$ is given as $b_j = \{b_1^j, b_2^j, \ldots, b_{k_j}^j\}$. The set $b = \{b_1, b_2, \ldots, b_n\}$ represents the set of bid vectors of the IoT devices for all the tasks. Based on the set $\boldsymbol{I}$, a non-conflict graph $\mathbb{G}(\mathcal{V}, \mathcal{E})$ is constructed; where $\mathcal{V}$ is the set of vertices representing the tasks. An edge $(i,\ j) \in \mathcal{E}$ between the tasks i and j represents the fact that the pair $(i,\ j)$ have at least one IoT device that is associated to both the tasks. Once the graph is constructed, next target is to place the tasks along with their respective IoT devices to different time slots so as to preserve the assumptions made. The set of time slots to which all the tasks are placed in, is given as $\tau = \{1, 2, \ldots, \kappa\}$; where κ is the number of time slots available. In *peer grading* phase, each IoT device $\mathbb{E}_i$ provides a rank list over the subset of IoT devices associated with task $\mathbb{T}_j$ denoted by $\succ_i^j$, where $\mathbb{E}_\ell \succ_i^j \mathbb{E}_k$ means that the IoT device $\mathbb{E}_i$ ranks $\mathbb{E}_\ell$ above $\mathbb{E}_k$. For each task $\mathbb{T}_j$, this *peer grading* process will result in the quality IoT devices. Now, the next target is to select the subset of IoT devices from the quality IoT devices for each task and decide their payment. The allocation vector for all the tasks is given as $\mathbb{A} = \{\mathbb{A}_1, \mathbb{A}_2, \ldots, \mathbb{A}_n\}$; where $\mathbb{A}_i$ contains the IoT devices selected for task $\mathbb{T}_i$. Similarly, the payment vector of all the IoT devices for n tasks is given as $\boldsymbol{P} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_n\}$. Here, $\boldsymbol{P}_j$ is the payment vector of IoT devices associated with task $\mathbb{T}_j$ and is given as $\boldsymbol{P}_j = \{\boldsymbol{P}_1^j, \ldots, \boldsymbol{P}_{k_j}^j\}$; where $\boldsymbol{P}_i^j$ is the payment received by IoT device $\mathbb{E}_i$ for executing task $\mathbb{T}_j$. The utility achieved by any i^{th} IoT device for each task $\mathbb{T}_j$ could be defined as the payment it received for executing task $\mathbb{T}_j$ minus the valuation of an IoT device for task $\mathbb{T}_j$ *i.e.* $u_i^j = \boldsymbol{P}_i^j - v_i^j$, if $\mathbb{E}_i$ is considered for task $\mathbb{T}_j$; otherwise 0.

Definition 1 (Incentive Compatible (IC) or Truthful Nisan et al. (2007)) A mechanism is said to be truthful, if reporting the true valuation by any agent i will maximize its utility irrespective of the valuations of other agents. In our case, for any arbitrary IoT device $\mathbb{E}_i$ for task $\mathbb{T}_j$ the utility relation is $u_i^j = \boldsymbol{P}_i^j - v_i^j \geq \boldsymbol{P}_i^j - b_i^j = \hat{u}_i^j$; where u_i^j is the utility when $\mathbb{E}_i$ reports true value and $\hat{u}_i^j$ is the utility when reporting the bid other than the true value $b_i^j \neq v_i^j$.

Definition 2 (Individual Rationality (IR) Nisan et al. (2007)) A mechanism is said to be individually rational if every agent i results in non-negative utility. More formally in our case, $u_i^j \geq 0$ when participating in the system.

Definition 3 (Budget Feasibility (BF) Singer (2010)) A mechanism is said to be budget feasible if the total payment made to the agents are within total budget. In our case, fix a task $\mathbb{T}_j$ we have, $\sum_{i=1}^{k_j} \boldsymbol{P}_i^j \leq \mathbb{B}_j$.

4 Proposed Mechanism: TUBE-TAP

In this section, we have proposed a *truthful* mechanism namely TUBE-TAP. The main components of the TUBE-TAP are: *Time slot allocation heuristic*, *Quality determination rule*, and *Allocation and payment rule.*

4.1 Time Slot Allocation Heuristic

The underlying idea behind proposing *Time slot allocation heuristic* motivated by[2] is to distribute the tasks into different time slots, so that: (a) the IoT devices gets the privilege to execute all the tasks for which they have shown their interest; (b) each IoT device executes a single task at a time.

4.1.1 Outline of Time slot allocation heuristic

Time slot allocation heuristic

First Phase:

1. Pick a task $\mathbb{T}_i$ which has less than κ adjacent tasks in a graph $\mathbb{G}$.
2. Put $\mathbb{T}_i$ on the stack and remove it along with the incident edges from the graph $\mathbb{G}$.
3. Repeat step 1 and 2, until the graph $\mathbb{G}$ is non-empty.

Second Phase: In each iteration:

1. Pop the task present at the top of the stack.
2. Assign it the lowest numbered time slot that is not assigned to any of its neighbouring tasks.

4.1.2 Detailed Time slot allocation heuristic

The first phase of the mechanism is depicted in line $2-9$ of Algorithm 1. In each iteration of *while* loop in line $2-9$, a task with neighbours less than κ

[2] https://www.youtube.com/watch?v=dJfQQNY7NdU

(κ time slots are available) is picked-up and is pushed into the stack S. Next, the recently pushed task is removed from the graph $\mathbb{G}$ along with its incident edges.

Algorithm 1: Time slot allocation heuristic ($\mathbb{G}$, κ)

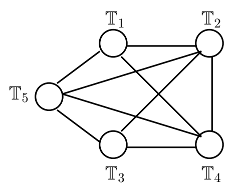
1 $\mathbb{G}' \leftarrow \mathbb{G}$, $S \leftarrow \phi$
2 **while** $\mathbb{G} \neq \phi$ **do**
3   **foreach** $\mathbb{T}_j \in \mathcal{V}$ **do**
4   **if** $|adj(\mathbb{T}_j)| < \kappa$ **then**
5   Push(S, $\mathbb{T}_j$) `// Task` $\mathbb{T}_j$ `is pushed into the stack` S
6             $\mathbb{G} \leftarrow \mathbb{G} \setminus \{\mathbb{T}_j\}$ `// Task` $\mathbb{T}_j$ `is removed from` $\mathbb{G}$
7   **end**
8   **end**
9 **end**
10 **while** $S \neq \phi$ **do**
11     $\Bbbk \leftarrow$ Pop(S) `//` $\Bbbk$ `holds an element popped-up from stack` S
12     $\mathbb{G} \leftarrow \mathbb{G} \cup \{\Bbbk\}$
13   Assign $\Bbbk$ the lowest numbered time slot that is not assigned to any of its neighbours.
14 **end**
15 **return** $\mathbb{G}$

In the second phase, shown in line $10-14$ of Algorithm 1, the actual process of time slot allocation is carried out. For each iteration of *while* loop in line 10-14, the top element is popped out of the stack S and held in $\Bbbk$. The element held in $\Bbbk$ is added back to graph $\mathbb{G}$. Each time a task is added in a graph $\mathbb{G}$ the information about the neighbouring tasks is fetched from $\mathbb{G}'$ graph. Now, the task added in current iteration is assigned a lowest numbered time slot that is not assigned to its neighbours using line 13. The *while* loop terminates once the stack is empty. Finally, in line 15 a graph $\mathbb{G}$ containing the information about the assigned time slot to each of the task is returned.

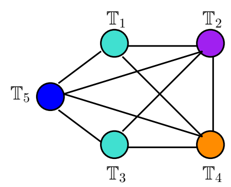
Example 1 For the understanding purpose, we have considered 5 tasks and 20 IoT devices.

Tasks	IoT Devices
$\mathbb{T}_1$	$\mathbb{E}_1$, $\mathbb{E}_3$, $\mathbb{E}_4$, $\mathbb{E}_6$, $\mathbb{E}_9$, $\mathbb{E}_{10}$, $\mathbb{E}_{13}$, $\mathbb{E}_{15}$, $\mathbb{E}_{17}$
$\mathbb{T}_2$	$\mathbb{E}_2$, $\mathbb{E}_3$, $\mathbb{E}_5$, $\mathbb{E}_6$, $\mathbb{E}_{10}$, $\mathbb{E}_{12}$, $\mathbb{E}_{16}$, $\mathbb{E}_{18}$
$\mathbb{T}_3$	$\mathbb{E}_{12}$, $\mathbb{E}_{16}$, $\mathbb{E}_{19}$
$\mathbb{T}_4$	$\mathbb{E}_1$, $\mathbb{E}_3$, $\mathbb{E}_4$, $\mathbb{E}_6$, $\mathbb{E}_7$, $\mathbb{E}_9$, $\mathbb{E}_{10}$, $\mathbb{E}_{19}$
$\mathbb{T}_5$	$\mathbb{E}_6$, $\mathbb{E}_7$, $\mathbb{E}_9$, $\mathbb{E}_{10}$, $\mathbb{E}_{11}$, $\mathbb{E}_{14}$, $\mathbb{E}_{15}$, $\mathbb{E}_{19}$, $\mathbb{E}_{20}$

(a) IoT devices interests over tasks



(b) Non-conflict graph



(c) Time slot allocation

Fig. 2: Detailed illustration of Algorithm 1

Let the budget associated with the 5 tasks are: $\mathbb{B}_1 = 50\$$, $\mathbb{B}_2 = 25\$$, $\mathbb{B}_3 = 30\$$, $\mathbb{B}_4 = 60\$$, and $\mathbb{B}_5 = 15\$$. For each task, the interested set of IoT devices is depicted in Fig. 2a. Fig. 2a will be read as, for task $\mathbb{T}_3$ the interested set of quality IoT devices are $\mathbb{E}_{12}$, $\mathbb{E}_{16}$, and $\mathbb{E}_{19}$. Based on the configuration shown in Fig. 2a, a graph $\mathbb{G}$ is formed as shown in Fig. 2b. Note that the tasks $\mathbb{T}_1$ and $\mathbb{T}_3$ do not share any common IoT devices so they do not have an edge between them. The result of which, they can be placed in the same time slot. In our case the tasks $\mathbb{T}_1$ and $\mathbb{T}_3$ belong to the same time slot, say time slot 1. Tasks $\mathbb{T}_2$, $\mathbb{T}_4$, and $\mathbb{T}_5$ share a common IoT devices so they have an edge between them and will be placed in three different time slots. Also, these tasks have an edge with $\mathbb{T}_1$ and $\mathbb{T}_3$ so they can not be placed in time slot 1. The tasks $\mathbb{T}_2$, $\mathbb{T}_4$, and $\mathbb{T}_5$ are placed in time slot 2, time slot 3, and time slot 4 respectively.

4.2 Quality Determination Mechanism

As the quality of the IoT devices are unknown, in this section a mechanism is proposed for determining the quality of the IoT devices. First, the outline of the *Quality determination mechanism* is presented in subsection 4.2.1 and in subsection 4.2.2 the detailed version of the mechanism is discussed.

4.2.1 Outline of The Quality Determination Mechanism

Quality determination mechanism

Repeat:

1. For each task T_i, assign r IoT devices to r' other IoT devices for the ranking purpose; here $r' \gg r$.
2. Select an IoT device that appears at first place in most of the rankings.

Until: Each IoT device is considered for the ranking.

4.2.2 Detailed Quality Determination Mechanism

Algorithm 2: Main routine ($\mathbb{G}$, $\mathbb{B}$, $\boldsymbol{I}$, τ, $\mathbb{T}$, b)

Output: $\mathbb{A}$, $\boldsymbol{P}$

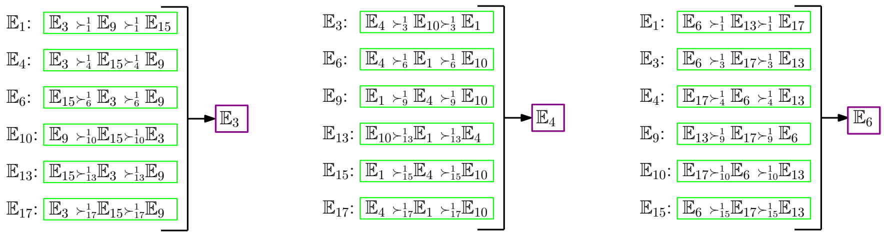
foreach $i \in \tau$ **do**
  foreach $\mathbb{T}_j \in i$ **do**
    $(\pi^j, \tilde{b}_j) \leftarrow$ Quality determination mechanism $(\mathbb{T}_j, \boldsymbol{I}^j)$
    $(\mathbb{A}'_j, \boldsymbol{P}'_j) \leftarrow$ Allocation and payment rule $(\pi^j, \tilde{b}_j, \mathbb{B}_j)$
    $\mathbb{A} \leftarrow \mathbb{A} \cup \mathbb{A}'_j$
    $\boldsymbol{P} \leftarrow \boldsymbol{P} \cup \boldsymbol{P}'_j$
  end
end
return $\mathbb{A}$, $\boldsymbol{P}$

The idea behind providing the *Main routine* is to capture each task of the system present in different time slots. In main routine, line $1-8$ keeps track of each time slot and in each time slot each task is taken care by line $2-7$. Line 9 returns the allocation and payment vectors for all the tasks in the system. In Algorithm 3, initializations are done in line 1. In line 2, Ψ_j' and Ψ_j keeps the copy of the IoT devices that execute the task $\mathbb{T}_j$. The *do while* loop in line 3-14 iterates until all the IoT devices are ranked up. Using line 7-10 the record about the top ranked IoT device by each $\mathbb{E}_i \in \varphi$ is kept in the $\mathcal{N}'$. In line 11, $\mathbf{\Phi}_j$ captures the IoT device that was ranked top by most of the IoT devices for task $\mathbb{T}_j$. Line 13 removes the IoT devices that are already ranked, from Ψ_j. Finally, line 15 returns $\mathbf{\Phi}_j$ that contains the quality IoT devices for task $\mathbb{T}_j$, and $\tilde{b}_j$.

Algorithm 3: Quality determination mechanism ($\mathbb{T}_j$, $\boldsymbol{I}^j$)

Output: $\mathbf{\Phi}_j \leftarrow \phi$
1 $\Psi \leftarrow \phi$, $\varphi \leftarrow \phi$, $\mathcal{N}' \leftarrow \phi$, $\beta \leftarrow \phi$
2 $\Psi_j' = \Psi_j = \boldsymbol{I}^j$
3 **do**
4 $\quad$ $\Psi \leftarrow$ Pick_random (Ψ_j, r) `// Pick` r `IoT devices from` Ψ_j`.`
5 $\quad$ $\varphi \leftarrow$ Pick_random ($\Psi_j' \setminus \Psi$, r') `// Pick` r' `IoT devices from` $\Psi_j' \setminus \Psi$`.`
6 $\quad$ Assign the completed task $\mathbb{T}_j$ of each IoT device in Ψ to the IoT devices in φ for ranking purpose.
7 $\quad$ **forall the** $\mathbb{E}_i \in \varphi$ **do**
8 $\quad\quad$ $\beta \leftarrow$ Select_best($\succ_i^j$) `// Select top ranked IoT device from` $\mathbb{E}_i's$ `ranked list for task` $\mathbb{T}_j$ `given as` $\succ_i^j$`.`
9 $\quad\quad$ $\mathcal{N}' \leftarrow \mathcal{N}' \cup \{\beta\}$ `//` $\mathcal{N}'$ `allows the duplication of elements.`
10 $\quad$ **end**
11 $\quad$ $\mathbf{\Phi}_j \leftarrow \mathbf{\Phi}_j \cup \{\max\limits_{\mathbb{E}_k \in \mathcal{N}'}\{|S_k|\}\}$ `//` S_k `is the set of` $E_k's$ `in` $\mathcal{N}'$`.`
12 $\quad$ $\tilde{b}_j \leftarrow \tilde{b}_j \cup \{b_k^j\}$ `//` $\tilde{b}_j$ `is the bid vector of quality IoT devices.`
13 $\quad$ $\Psi_j \leftarrow \Psi_j \setminus \Psi$
14 **while** $\Psi_j \neq \phi$
15 **return** $\mathbf{\Phi}_j$, $\tilde{b}_j$

Example 2 For the detailed illustration of Algorithm 3 we have considered the set-up discussed in Example 1. In this example, we have illustrated Algorithm 3 for one task, say task $\mathbb{T}_1$. However, similar procedure could be followed for the remaining tasks. For the 1^{st} iteration of the peer grading (PG) process, we have randomly selected 3 IoT devices ($r = 3$) say $\mathbb{E}_3$, $\mathbb{E}_9$, and $\mathbb{E}_{15}$ and assigned their executed task to the remaining IoT devices for the reviewing purposes. Next, following Algorithm 3, we have to check which IoT device among $\mathbb{E}_3$, $\mathbb{E}_9$, and $\mathbb{E}_{15}$ has been top ranked by the majority of the peers. From Fig. 3a one can see that $\mathbb{E}_3$ has been top ranked by the majority of the IoT devices. So, we have $\mathbf{\Phi}_1 = \{\mathbb{E}_3\}$.



(a) PG (1^{st} iteration) (b) PG (2^{nd} iteration) (c) PG (3^{rd} iteration)

Fig. 3: Detailed illustration of Algorithm 3

In the similar fashion, we can continue with the other iterations of the peer grading process as shown in Fig. 3b and Fig. 3c and determine the quality IoT devices. At the end of the peer grading process, we get $\mathbf{\Phi}_1 = \{\mathbb{E}_3, \mathbb{E}_4, \mathbb{E}_6\}$.

4.3 Allocation and Payment Rule

This section explains the *Allocation and payment rule* presented in the Algorithm 4.

Algorithm 4: Allocation and payment rule (π^j, $\tilde{b}_j$, $\mathbb{B}_j$)

Output: $\mathbb{A}_j$, $\boldsymbol{P}_j$

`/* Allocation Rule */`

1 Sort(π^j, $\tilde{b}_j$) `// Sort` π^j `based on` $\tilde{b}_j$ `as` $b_1^j \le b_2^j \le \ldots \le b_{\tilde{k}_j}^j$ `; such that` $\tilde{k}_j < k_j$

2 $k \leftarrow 1$

3 **while** $b_i^j \le \frac{\mathbb{B}_j}{k}$ **do**

4 $\quad$ $\mathbb{A}_j \leftarrow \mathbb{A}_j \cup \{\mathbb{E}_i\}$ `//` $\mathbb{A}_j$ `keeps track of winning IoT devices.`

5 $\quad$ $k \leftarrow k + 1$

6 **end**

`/* Payment Rule */`

7 **foreach** $\mathbb{E}_i \in \mathbb{A}_j$ **do**

8 $\quad$ $\boldsymbol{P}_i^j \leftarrow \{min\{\frac{\mathbb{B}_j}{k},\ b_{k+1}^j\}\}$

9 $\quad$ $\boldsymbol{P}_j \leftarrow \boldsymbol{P}_j \cup \{\boldsymbol{P}_i^j\}$

10 **end**

11 **return** $\mathbb{A}_j$, $\boldsymbol{P}_j$

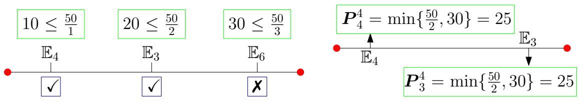
Considering the allocation rule, in line 1 first the quality IoT devices in π^j is sorted in increasing order based on the bid vector $\tilde{b}_j$. The variable k is initialized to 1. The *while* loop in line $3 - 6$ determines the largest index

k that satisfies the stopping condition of the *while* loop. Talking about the payment rule, for each $\mathbb{E}_i$ in $\mathbb{A}_j$ the minimum among $\frac{\mathbb{B}_j}{k}$ and b^j_{k+1} is taken as the payment. Finally, line 11 returns the allocation and payment for task $\mathbb{T}_j$.

Example 3 For understanding the allocation and payment rule, let us continue with the quality IoT devices resulted from Example 2. The budget given for task $\mathbb{T}_1$ is 50\$. The quality IoT devices along with their bid values are depicted in Fig. 4a. Utilizing Algorithm 4 in the set-up shown in Fig. 4a. First the IoT devices are sorted in increasing order of their bid value as shown in Fig. 4b. Next, from the sorted ordering, first $\mathbb{E}_4$ is picked up and is considered, as the check $10 \leq \frac{50}{1}$ is satisfied for $\mathbb{E}_4$. Next, $\mathbb{E}_3$ is picked up from the ordering and is also considered because of the similar reason. Next, $\mathbb{E}_6$ is picked up from the ordering and will be not be considered as the check $30 \leq \frac{50}{3}$ is not satisfied.

IoT Devices	$\mathbb{E}_3$	$\mathbb{E}_4$	$\mathbb{E}_6$
Bid values	20	10	30

(a) Bid value configuration



(b) Allocation resulted (c) Payment determination

Fig. 4: Detailed illustration of Algorithm 4.

So, we have $\mathbb{A}_1 = \{\mathbb{E}_4, \mathbb{E}_3\}$ as the winning set. We get the k value as 2 for our example. Next, the payment calculation of $\mathbb{E}_4$ and $\mathbb{E}_3$ is presented in Fig. 4c. For $\mathbb{E}_4$ we have $\boldsymbol{P}^1_4 = min\{\frac{50}{2},\ 30\} = 25$, and for $\mathbb{E}_3$ we have $\boldsymbol{P}^1_3 = min\{\frac{50}{2},\ 30\} = 25$. In this case, it can be seen that the total payment made to the IoT devices is equal to budget *i.e.* 50. For $\mathbb{E}_4$ the utility is $u^1_4 = \boldsymbol{P}^1_4 - v^1_4 = 25 - 10 = 15$ and for $\mathbb{E}_3$ the utility is $u^1_3 = \boldsymbol{P}^1_3 - v^1_3 = 25 - 20 = 5$. It can be seen that the utility of the IoT devices are non-negative, so the TUBE-TAP mechanism is individually rational.

In order to see the *truthful* behaviour of TUBE-TAP, let us suppose that the IoT device $\mathbb{E}_6$ reports its bid value as 15 instead of 30. In such scenario, the allocation rule of TUBE-TAP mechanism will give $\mathbb{A}_1 = \{\mathbb{E}_4, \mathbb{E}_6\}$ as the winning set. For $\mathbb{E}_4$ we have the payment as $\boldsymbol{P}^1_4 = min\{\frac{50}{2},\ 20\} = 20$, and for $\mathbb{E}_6$ we have the payment as $\boldsymbol{P}^1_6 = min\{\frac{50}{2},\ 20\} = 20$. For $\mathbb{E}_4$ the utility is $u^1_4 = \boldsymbol{P}^1_4 - v^1_4 = 20 - 10 = 10$ and for $\mathbb{E}_6$ the utility is $u^1_6 = \boldsymbol{P}^1_6 - v^1_6 = 20 - 30 = -10$. So, by misreporting its true value the IoT device $\mathbb{E}_6$ is not gaining. Hence, TUBE-TAP is truthful.

5 Analysis of TUBE-TAP

This section presents the analysis of TUBE-TAP.

Proposition 1 *The mechanism in Singer (2010) has an approximation ratio of 2.*

Lemma 1 *TUBE-TAP is truthful.*

Proof The proof is divided into two cases. In the first case, we have taken an arbitrary winning IoT device into consideration and discuss the impact on its gain (or utility), when it *deviates* from its true valuation. In second case, we have considered any arbitrary losing IoT device and analysis similar to Case 1 is done. Fix a task $\mathbb{T}_j$.

Case 1 Let us suppose that i^{th} winning IoT device deviates from its true value and reports a bid value $b_i^j < v_i^j$. As the IoT device $\mathbb{E}_i$ was winning with v_i^j, it will continue to win with b_i^j because by reporting value lesser than the true value, it will be appearing early in the ordering. So, its utility will be $\hat{u}_i^j = \boldsymbol{P}_i^j - v_i^j$ which is same as u_i^j. But, if it reports $b_i^j > v_i^j$, this gives rise to two possibilities. One possibility could be, it would continue to win by appearing later in the ordering and in that case its utility will be $\hat{u}_i^j = \boldsymbol{P}_i^j - v_i^j = u_i^j$. Another possibility could be, it may lose by appearing later in the ordering in that case its utility will be $\hat{u}_i^j = 0$.

Case 2 Let us suppose that i^{th} losing IoT device deviates from its true value and reports a bid value $b_i^j > v_i^j$. As the IoT device $\mathbb{E}_i$ was losing with v_i^j, it will continue to lose by b_i^j because by deviating this way it will be appearing later in the ordering. So, its gain will be $\hat{u}_i^j = 0$ which is same as u_i^j. But, if it reports $b_i^j < v_i^j$, then the two possibilities arises. One possibility could be, by deviating this way it could appear early in the ordering but still continue to lose and in that case $\hat{u}_i^j = 0$ which is same as u_i^j. Another possibility could be, it could win, in that case it has defeated the IoT device $\mathbb{E}_k$ with valuation $v_k^j < v_i^j$ and hence $b_i^j < v_k^j$. In this case, its payment will be less as compared to its true valuation. So, its utility $\hat{u}_i^j = \boldsymbol{P}_i^j - v_i^j < 0$. Hence, no gain is achieved.

Considering *Case* 1 and *Case* 2, it can be concluded that the IoT devices cannot gain by misreporting their true value. So, TUBE-TAP is truthful.

Lemma 2 *In TUBE-TAP, for each task requester $\mathbb{R}_j$ the total payment $\boldsymbol{P}_j$ made to the IoT devices are within available budget $\mathbb{B}_j$. More formally, $\boldsymbol{P}_j = \sum\limits_{\mathbb{E}_i \in \mathbb{A}_j} \boldsymbol{P}_i^j \leq \mathbb{B}_j$. Also, $\sum\limits_{\mathbb{A}_j \in \mathbb{A}} \sum\limits_{\mathbb{E}_i \in \mathbb{A}_j} \boldsymbol{P}_i^j \leq \sum\limits_{\mathbb{T}_j \in \mathbb{T}} \mathbb{B}_j$.*

Proof The proof is presented in Appendix A.1.

Lemma 3 *The allocation resulted by TUBE-TAP is at most 2 allocation away from the optimal one i.e. $OPT \leq 2 \times OM$; where OPT is the optimal allocation and OM is the allocation resulted by TUBE-TAP.*

Proof Fix a task requester $\mathbb{R}_i$ and task $\mathbb{T}_i$. Let us suppose for the sake of contradiction that the OPT consists of k IoT devices *i.e.* $|OPT| = k$ and OM consists of less than $\frac{k}{2}$ IoT devices *i.e.* $|OM| < \frac{k}{2}$. It implies that, $b^i_{\frac{k}{2}} > \frac{\mathbb{B}_i}{k/2}$. Note, however that this is impossible since we assume that $b^i_{\frac{k}{2}} \leq \ldots \leq b^i_k$, and $\sum_{j=\frac{k}{2}}^{k} b_j^i \leq \mathbb{B}_i$, which implies that $b^i_{\frac{k}{2}} \leq \frac{\mathbb{B}_i}{k/2}$. Hence a contradiction.

Lemma 4 *Let $\mathbb{U}$ be the event given as $\mathbb{U} = \{\mathbb{E}_i$ is considered for task $\mathbb{T}_j\}$ and X_j^i is an indicator random variable defined as $X_j^i = I\{\mathbb{U}\}$. Then, the expectation is just the probability of the corresponding event i.e. $E[X_j^i] = Pr\{\mathbb{U}\}$.*

Proof The proof is presented in Appendix A.2.

Lemma 5 *The expected number of times any arbitrary $\mathbb{E}_i$ is considered (or winning) is given as $p \cdot k_i$; where k_i is the number of tasks for which the i^{th} IoT device has shown interest and p is the probability with which $\mathbb{E}_i$ is considered for a task. In other words, $E[X^i] = p \cdot k_i$; where X^i is the random variable measuring the number of times $\mathbb{E}_i$ is considered out of k_i.*

Proof Fix an IoT device $\mathbb{E}_i$, we now wish to compute the expected number of times the $\mathbb{E}_i$ is considered. We capture the total number of times $\mathbb{E}_i$ is considered out of k_i by X^i random variable. So, the expected number of times $\mathbb{E}_i$ is considered is given as $E[X^i]$. Our sample space for $\mathbb{E}_i$ IoT device for any task $\mathbb{T}_j$ is S= {$\mathbb{E}_i$ is considered for task $\mathbb{T}_j$, $\mathbb{E}_i$ not considered for task $\mathbb{T}_j$}. So, we have Pr{$\mathbb{E}_i$ is considered for task $\mathbb{T}_j$}= p and Pr{$\mathbb{E}_i$ is not considered for task $\mathbb{T}_j$} $= 1 - p$.

We define the indicator random variable X_j^i as X_j^i = I{$\mathbb{E}_i$ is considered for task $\mathbb{T}_j$}; where

$$X_j^i = \begin{cases} 1, & \textit{if } \mathbb{E}_i \textit{ is considered for task } \mathbb{T}_j \\ 0, & \textit{Otherwise} \end{cases} \tag{1}$$

The expected number of times $\mathbb{E}_i$ is considered for task $\mathbb{T}_j$ is simply the expected value of our indicator random variable X_j^i:

$$E[X_j^i] = E[I\{\mathbb{E}_i \ is\ considered\ for\ task\ \mathbb{T}_j\}]$$

As always with the indicator random variable, the expectation is just the probability of the corresponding event (using lemma 4):

$$E[X_j^i] = 1 \cdot Pr\{X_j^i = 1\} + 0 \cdot Pr\{X_j^i = 0\} = 1 \cdot p + 0 \cdot (1 - p) = 1 \cdot p = p$$

Now, let us consider the random variable that we are interested in and is given by $X^i = \sum_{j=1}^{k_i} X_j^i$. We can compute $E[X^i]$ by taking expectation both side and by linearity of expectation, we get:

$$E[X^i] = E\left[\sum_{j=1}^{k_i} X_j^i\right] = \sum_{j=1}^{k_i} E[X_j^i]$$

From lemma 4 it can be seen that, the expected value of any random variable is equal to the probability of the corresponding event. So,

$$E[X^i] = \sum_{j=1}^{k_i} Pr\{\mathbb{E}_i \ is\ considered\ for\ task\ \mathbb{T}_j\} = \sum_{j=1}^{k_i} p = p \cdot k_i$$

Hence, the claim survived. It is to be noted that if $p = \frac{1}{2}$, then the value of $E[X^i]$ boils down to $\frac{k_i}{2}$. It means that, any arbitrary $\mathbb{E}_i$ in expectation will be considered for half of number of tasks on which it has shown interest.

Lemma 6 *For any arbitrary IoT device $\mathbb{E}_i$ the expected number of longest contiguous rejection out of k_i tasks after which the IoT device is considered is given as $\Theta(\log_p k_i)$. More formally, we can say $E[Y] = \Theta(\log_p k_i)$; where Y is a random variable that captures the longest continuous rejection of any IoT device.*

Proof The proof is presented in Appendix A.3.

Lemma 7 *In our system, the probability that any arbitrary IoT device $\mathbb{E}_i$ is considered (or wins) for at least one time out of k_i is greater than or equal to $1 - \frac{1}{e^{p \cdot k_i}}$; where k_i is the number of tasks for which the i^{th} IoT device has shown interest. In other words, $Pr[X^i \geq 1] \geq \left(1 - \frac{1}{e^{p \cdot k_i}}\right)$; where X^i is the random variable measuring the number of times $\mathbb{E}_i$ IoT device is considered out of k_i.*

Proof The proof is presented in Appendix A.4.

6 Modeling in Divisible Setting

In this section, we provide the variant of our proposed model discussed in Section 3. Here, it is considered that, a distinct single task that is held by each task requester are divisible in nature. Other than this, there are some other constraints that are taken into consideration that helped making the system more *realistic*. It is considered that: 1) each task $\mathbb{T}_i$ has an estimated time t_i that is required for its completion. The estimated time vector of n available tasks is given as $t = \{t_1, t_2, \ldots, t_n\}$; 2) for each IoT device $\mathbb{E}_j$ we have a battery drainage time and is denoted by $\tilde{t}_j$. It means that the time by which its battery power will be drained out completely. For all the m IoT devices, it is given by the set $\tilde{t} = \{\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_m\}$; and 3) each task will be executed by the IoT devices in a collaborative manner.

The underlying solution approach that is utilized is, first, similar to the model discussed in Section 3 the tasks along with the IoT devices are distributed in different time slots. Next, in each time slot, for each task $\mathbb{T}_j$, the *peer grading* mechanism (discussed in Algorithm 3) is followed. Next, the task $\mathbb{T}_j$ is divided into constant number of heterogeneous sub-tasks *say* $\hat{k}_j$ given as $\mathbb{T}_j = \{\mathbb{T}_j^1, \mathbb{T}_j^2, \ldots, \mathbb{T}_j^{\hat{k}_j}\}$, where $\mathbb{T}_j^i$ is the i^{th} sub-task of task $\mathbb{T}_j$. By *heterogeneity*, we mean that the estimated completion time of the sub-tasks may differ among themselves. The estimated time of any sub-task $\mathbb{T}_j^i$ is given as t_j^i. The set $t^j = \{t_j^1, t_j^2, \ldots, t_j^{\hat{k}_j}\}$ represents the estimated time of all the sub-tasks of task $\mathbb{T}_j$. Now the questions are: a) *how the quality IoT devices that were associated with a task $\mathbb{T}_j$ will be distributed to these $\hat{k}_j$ heterogeneous sub-tasks?*,

b) *what payment policy is to be followed in such setting?* For this purpose, solution approach discussed in Algorithm 5 is utilized. For each sub-task, the assigned IoT devices forms a grand coalition and works collaboratively to complete the same. For the time being, it is assumed that for each sub-task we have limited (or constant) number of IoT devices and these IoT devices in coalition are capable of performing the sub-task successfully in the collaborative fashion. In our discussed set-up, the collaborative work culture is achieved as: each sub-task is divided into independent chunks and each chunk is given to the IoT device for the execution purpose. In order to do the division of sub-task among the members of the coalition, we have utilized the *Shapley value* concept. The underlying ideas behind utilizing the *Shapley value* concept for this purpose are: 1) the fair division of a sub-task could be achieved among the members of the grand coalition, and 2) it satisfies several useful properties such as *symmetry*, *null player property*, and *additivity* (defined in subsection 6.1) that is needed for the real time implementation of the system in true sense. After the execution, the result of each chunk is submitted and combined together to get the result of the sub-task. Now, using *Shapley value* concept, we distribute the available budget for the sub-task under consideration fairly among the available IoT devices in the coalition as their payment. The result of each sub-task is combined to get the final executed task. It is to be noted that, we have assumed that no IoT device is capable of completing a sub-task alone. In this set-up, one scenario that is deliberately overlooked is, instead of having the substantial number of IoT devices, the sub-tasks may have huge number of such devices those works collaboratively to accomplish the sub-task. This deliberation is due to the reason that, in literature it is pointed out that calculating the *Shapley value* for such huge set of IoT devices in a coalition while preserving its properties is NP-hard[3]. In the upcoming section (Section 7), we have relaxed this constraint and provided a *non-truthful budget feasible mechanism* to the enhanced version of the problem that utilizes the concept of *proportional share mechanism*.

It is to be noted that, due to involvement of IoT devices in a collaborative work culture, we have modelled the problem under investigation by utilizing the concept of *coalitional game theory* (CGT) Maschler et al. (2013). Defining a coalitional game for a sub-task of task under consideration. Let the set of IoT devices associated with the sub-task under consideration is given as $\mathcal{N}$. A coalitional game is a pair $(\mathcal{N}, v)$ where; $v : 2^{\mathcal{N}} \longmapsto \mathbb{R}$ associates with each coalition $\mathbb{F} \subseteq \mathcal{N}$ a real-valued pay-off $v(\mathbb{F})$ that the coalition's members (IoT devices) can distribute among themselves. We assume that $v(\phi) = 0$. Given a coalitional game $(\mathcal{N}, v)$, a single-valued solution concept $\boldsymbol{\nabla} : \mathcal{N} \times \mathbb{R}^{2^{|\mathcal{N}|}} \longmapsto \mathbb{R}^{|\mathcal{N}|}$ is a function mapping a coalitional game (that is, a set of IoT devices $\mathcal{N}$ and a value function v) to a vector of $|\mathcal{N}|$ real values, and let $\boldsymbol{\nabla}_i(\mathcal{N}, v)$ denote the i^{th} such real value (value of i^{th} IoT device in $(\mathcal{N}, v)$ according to $\boldsymbol{\nabla}$).

Given the above discussed set-up, our objective is to select a subset of IoT

[3] It could be tackled by utilizing the concept of Approximate Shapley value Fatima et al. (2008) that is reserved for our future work.

devices for each task in such a way that the required estimated time for each task could be attained by the remaining battery power of the IoT devices, working collaboratively. Along with this, it should also be ensured that the total payment made to the selected IoT devices for each task is within the allotted quota of budget while attaining a threshold quality. In this paper, for this set-up we have designed a ***no**n-**t**ruthful b**u**dget feasi**b**le m**e**chanism* (No-TUBE) motivated by Maschler et al. (2013); Shapley (1953) that allow us to attain the objective that we are interested in.

Definition 4 Given a coalitional game $(\mathcal{N},\ v)$, the Shapley value of i^{th} IoT device is given by

$$\boldsymbol{\nabla}_i(\mathcal{N}, v) = \frac{1}{|\mathcal{N}|!} \sum_{\mathbb{F} \subseteq \mathcal{N} \setminus \{i\}} |\mathbb{F}|!(|\mathcal{N}| - |\mathbb{F}| - 1)! \Big[v(\mathbb{F} \cup \{i\}) - v(\mathbb{F}) \Big] \tag{2}$$

where, the term $\Big[v(\mathbb{F} \cup \{i\}) - v(\mathbb{F}) \Big]$ is the marginal contribution of i^{th} IoT device when added to any coalition $\mathbb{F}$ that does not contain i.

6.1 Several Useful Properties

In this section, we discuss the properties satisfied by the *Shapley value* Maschler et al. (2013); Shapley (1953):

Definition 5 (Symmetry) Let $(\mathcal{N}, v)$ be a coalitional game, and let $\mathcal{N}_i$, $\mathcal{N}_j \in \mathcal{N}$. Then, we can say the IoT devices $\mathcal{N}_i$ and $\mathcal{N}_j$ are symmetric if for every coalition $\mathbb{F} \subseteq \mathcal{N} \setminus \{\mathcal{N}_i,\ \mathcal{N}_j\}$ they always contribute the same amount. More formally,

$$v(\mathbb{F} \cup \{\mathcal{N}_i\}) = v(\mathbb{F} \cup \{\mathcal{N}_j\}) \tag{3}$$

A solution concept $\boldsymbol{\nabla}$ satisfies symmetry if for every coalitional game $(\mathcal{N}, v)$ and every pair of symmetric IoT devices $\mathcal{N}_i$ and $\mathcal{N}_j$ in the game, we have:

$$\boldsymbol{\nabla}_i(\mathcal{N}, v) = \boldsymbol{\nabla}_j(\mathcal{N}, v) \tag{4}$$

Definition 6 (Dummy IoT device) An IoT device $\mathcal{N}_i$ is called a dummy IoT device in a coalitional game $(\mathcal{N},\ v)$, if for every coalition $\mathbb{F} \subseteq \mathcal{N}$, including the empty coalition, the value that $\mathcal{N}_i$ contributes is 0. More formally,

$$v(\mathbb{F}) = v(\mathbb{F} \cup \{\mathcal{N}_i\}) \tag{5}$$

For any v, if $\mathcal{N}_i$ is a dummy IoT device in a coalitional game $(\mathcal{N},\ v)$ then $\boldsymbol{\nabla}_i$ is given as:

$$\boldsymbol{\nabla}_i(\mathcal{N}, v) = 0 \tag{6}$$

Definition 7 (Additivity) For the the game ($\mathcal{N}$, $v_i + v_j$) with ($\mathcal{N}$, v_i) and ($\mathcal{N}$, v_2) are the two separate coalitional game, then for every coalition $\mathbb{F} \subseteq \mathcal{N}$ we can have,

$$(v_i + v_j)(\mathbb{F}) = v_i(\mathbb{F}) + v_j(\mathbb{F}) \tag{7}$$

A solution concept $\boldsymbol{\nabla}_i$ for each $\mathcal{N}_i$ satisfies the additivity property if for every pair of coalitional game ($\mathcal{N}$, v_i) and ($\mathcal{N}$, v_j);

$$\boldsymbol{\nabla}_i(\mathcal{N}, v_i + v_j) = \boldsymbol{\nabla}_i(\mathcal{N}, v_i) + \boldsymbol{\nabla}_i(\mathcal{N}, v_j) \tag{8}$$

6.2 Proposed Mechanism: NoTUBE

In this section, we propose a ***no****n-**t**ruthful b**u**dget feasi**b**le m**e**chanism* (NoTUBE) for our setting. First, the outline of the NoTUBE is given in sub section 6.3. The detailing of the NoTUBE is illustrated in sub section 6.4.

6.3 Outline of NoTUBE

In this section, the outline of NoTUBE is presented.

NoTUBE

1. Follow Algorithm 1 and 3 for distributing the tasks into different time slots and for determining the quality IoT devices respectively.
2. For each task, sort the quality IoT devices in decreasing order of their battery power left.
3. In each time slot, divide each task into constant number of heterogeneous sub-tasks.
4. From the sorted ordering, assign an IoT device to the sub-task having the highest remaining estimated time until each sub-task has substantial number of IoT devices.
5. For each sub-task, utilize the *Shapley value* concept for distributing the part of sub-task and the part of budget as payment among the member of the coalition in a fair way.

6.4 Detailing of NoTUBE

This section explains the detailing of the NoTUBE presented in Algorithm 5. For each iteration of *for* loop in line 1-28, a time slot is considered. In each time slot the available tasks are taken care by line 2-27. Line 3 sorts the quality IoT devices associated with task $\mathbb{T}_j$ in decreasing order of battery drainage time. After that, the task $\mathbb{T}_j$ is divided into heterogeneous sub-tasks as stated in line 4. Line 5-13 depicts the coalition formation procedure for all the sub-tasks of a task under consideration. Line 6 selects an IoT device from the sorted ordering and held in ℓ^*. In line 7, the sub-task with highest remaining estimated time is selected and stored in q^*.

Algorithm 5: NoTUBE ($\mathbb{T}$, $\mathbb{B}$, $\tilde{t}$, t, τ)

Output: $\mathcal{T} \leftarrow \phi$, $\mathbb{A} \leftarrow \phi$, $\boldsymbol{P} \leftarrow \phi$

1 **foreach** $i \in \tau$ **do**
2   **foreach** $\mathbb{T}_j \in i$ **do**
3     $\mathbb{L} \leftarrow \text{Sort}(\pi^j)$ `//` π^j `is the set of quality IoT devices for` $\mathbb{T}_j$`.`
4   Divide the task $\mathbb{T}_j$ into $\hat{k}_j$ sub-tasks *i.e.* $\mathbb{T}_j = \{\mathbb{T}_j^1, \mathbb{T}_j^2, \ldots, \mathbb{T}_j^{\hat{k}_j}\}$
5   **while** $t_j^i > k^*$ *for any* $\mathbb{T}_j^i \in \mathbb{T}_j$ **do**
6       $\ell^* \leftarrow \text{Select }(\mathbb{L})$
7       $q^* \leftarrow argmax_{\mathbb{T}_j^i \in \mathbb{T}_j}\{t^j\}$
8   Assign(q^*, ℓ^*) `// assigns the IoT device in` ℓ^* `to the sub-task in` q^*`.`
9       $\mathcal{N}_j^i \leftarrow \mathcal{N}_j^i \cup \{\ell^*\}$
10       $t_j^i = t_j^i$ - $\tilde{t}_*$ `//` $\tilde{t}_*$ `is the battery drainage time of IoT device in` ℓ^*`.`
11   Update(t^j) `// update the estimated time vector of` $\mathbb{T}_j$
12       $\mathbb{L} \leftarrow \mathbb{L} \setminus \{\ell^*\}$
13   **end**
14   **foreach** $\mathbb{T}_j^i \in \mathbb{T}_j$ **do**
15   **foreach** $\mathbb{E}_\ell \in \mathcal{N}_j^i$ **do**
16         $(\boldsymbol{\nabla}'_\ell(\mathcal{N}_j^i, v),\ \boldsymbol{\nabla}_\ell(\mathcal{N}_j^i, v)) = \frac{1}{|\mathcal{N}_j^i|!} \sum_{\mathbb{F}_j^i \subseteq \mathcal{N}_j^i \setminus \{\ell\}} |\mathbb{F}_j^i|!(|\mathcal{N}_j^i| - |\mathbb{F}_j^i| - 1)! \Big[v(\mathbb{F}_j^i \cup \{\ell\}) - v(\mathbb{F}_j^i) \Big]$
17         $\mathcal{T}_\ell^j \leftarrow \mathcal{T}_\ell^j \cup \{\boldsymbol{\nabla}'_\ell(\mathcal{N}_j^i, v)\}$
18         $\boldsymbol{P}_\ell^j \leftarrow \boldsymbol{P}_\ell^j \cup \{\boldsymbol{\nabla}_\ell(\mathcal{N}_j^i, v)\}$
19   **end**
20       $\mathcal{T}_j \leftarrow \mathcal{T}_j \cup \mathcal{T}_\ell^j$
21       $\mathbb{A}_j \leftarrow \pi^j$
22       $\boldsymbol{P}_j \leftarrow \boldsymbol{P}_j \cup \boldsymbol{P}_\ell^j$
23   **end**
24     $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_j$
25     $\mathbb{A} \leftarrow \mathbb{A} \cup \mathbb{A}_j$
26     $\boldsymbol{P} \leftarrow \boldsymbol{P} \cup \boldsymbol{P}_j$
27   **end**
28 **end**
29 **return** $\mathcal{T}$, $\mathbb{A}$, $\boldsymbol{P}$

$\mathcal{N}_j^i$ maintains the set of IoT devices allocated to $\mathbb{T}_j^i$ using line 9. In line 10, $\tilde{t}_*$ is subtracted from the estimated time of the sub-task $\mathbb{T}_j^i$. The *while* loop terminates only when, for all the sub-tasks the remaining estimated time is greater than some small negative constant k^*. By taking k^* this way (say for example k^* = -5), we guarantee that each of the sub-task will be having a substantial number of IoT devices. Once the coalition for each sub-task is determined, next objective is to have a fair division of each sub-task as chunks and the budget as payment among the members of the coalition. Utilizing, line 14-23 the required objective is achieved. In line 17, $\mathcal{T}_\ell^j$ is the estimated time of chunk of the sub-task $\mathbb{T}_j^i$ allotted to $\mathbb{E}_\ell$. $\boldsymbol{P}_\ell^j$ is the payment of IoT device associated with task $\mathbb{T}_j$ calculated in line 18. In line 24, $\mathcal{T}$ keeps track of amount of time IoT devices in $\mathbb{A}$ execute the allocated task. The allocation vector and payment vector of all the winning IoT devices are held in $\mathbb{A}$ and $\boldsymbol{P}$ respectively as depicted in line 25 and 26. Line 29 returns the outputs.

6.5 Illustrative Example

In this section, we provide the working example for showing the *Shapley value* calculation in action in Algorithm 5. Consider a task $\mathbb{T}_r$ as the translation of a book "*Introduction to Algorithms*" by *Cormen et al.* written in English into 5 different languages (such as *French*, *Spanish*, *Chinese*, *Japanese*, and *German*). The translation process is carried out in the Lab of an Institute equipped with say 20 quality IoT devices given as $\mathbb{E}_1, \mathbb{E}_2, \ldots, \mathbb{E}_{20}$. These IoT devices are programmed in a way that they are capable of translating any given language to other languages. The estimated completion time of the task $\mathbb{T}_r$ is given as 15 hrs. In our running example, translating a book written in English language to some other language is considered as a sub-task. It means that, translating a book "*Introduction to Algorithms*" by *Cormen et al.* written in English to say French is considered as a sub-task and is represented as $\mathbb{T}_r^1$ in our case. In the similar fashion, from English to Spanish as $\mathbb{T}_r^2$, English to Chinese as $\mathbb{T}_r^3$, English to Japanese as $\mathbb{T}_r^4$, and English to German as $\mathbb{T}_r^5$. The estimated completion time of the sub-tasks $\mathbb{T}_r^1$, $\mathbb{T}_r^2$, $\mathbb{T}_r^3$, $\mathbb{T}_r^4$, and $\mathbb{T}_r^5$ are 5hrs, 4hrs, 3hrs, 2hrs, and 1hr respectively. The total budget associated with the task $\mathbb{T}_r$ is 120\$. The total budget associated with task $\mathbb{T}_r$ will be distributed among the sub-tasks in the proportion of their estimated completion time. It means that, the total budget of 120\$ associated with the task $\mathbb{T}_r$ will be distributed among the sub-tasks $\mathbb{T}_r^1$, $\mathbb{T}_r^2$, $\mathbb{T}_r^3$, $\mathbb{T}_r^4$, and $\mathbb{T}_r^5$ in the ratio 5:4:3:2:1. So, the budget associated with the sub-tasks $\mathbb{T}_r^1$, $\mathbb{T}_r^2$, $\mathbb{T}_r^3$, $\mathbb{T}_r^4$, and $\mathbb{T}_r^5$ is 40\$, 32\$, 24\$, 16\$, and 8\$ respectively. Now, let us consider a sub-task $\mathbb{T}_r^2$ (*i.e.* translating a book "*Introduction to Algorithms*" by *Cormen et al.* written in *English* to *Spanish*) to show the *Shapley value* calculation. Let the IoT devices that are assigned to the sub-task $\mathbb{T}_r^2$ be $\mathbb{E}_2$, $\mathbb{E}_3$, $\mathbb{E}_{15}$, and $\mathbb{E}_7$. The battery drainage time for these IoT devices is given as: $\tilde{t}_2 = 2hrs$, $\tilde{t}_3 = 1hr$, $\tilde{t}_{15} = 1hr$, and $\tilde{t}_7 = 1hr$. By forming the grand coalition, the allocated IoT devices are required to achieve the estimated time of 4hrs for task $\mathbb{T}_r^2$. Now, in the grand coalition, the fraction of sub-task $\mathbb{T}_r^2$ and the fraction of budget associated with $\mathbb{T}_r^2$ given to the IoT devices $\mathbb{E}_2$, $\mathbb{E}_3$, $\mathbb{E}_{15}$, and $\mathbb{E}_7$ are calculated as: (Here, ${\xi'}_k^i$ and ξ_k^i represents the i^{th} term in the *Shapley value* calculation for the fair distribution of the sub-task and the budget to the k^{th} member of the coalition respectively.)

– **Case 1:** (When coalition $\mathbb{F}_r^2$ is empty *i.e.* $|\mathbb{F}_r^2| = 0$ and then $\mathbb{E}_2$ is added in $\mathbb{F}_r^2$.)

$$\xi'^1_2 = \xi_2^1 = \frac{0! \times (4-0-1)!}{4!} \times [v(\{\mathbb{E}_2\}) - v(\phi)] = \frac{0! \times (4-0-1)!}{4!} \times [0-0] = 0$$

– **Case 2:** (When coalition $\mathbb{F}_r^2$ contains 1 member *i.e.* $|\mathbb{F}_r^2| = 1$ and then $\mathbb{E}_2$ is added in $\mathbb{F}_r^2$.)

$$\xi'^2_2 = \xi_2^2 = 3\times\frac{1! \times (4-1-1)!}{4!} \times [v(X) - v(Y)] = 3\times\frac{1! \times (4-1-1)!}{4!} \times [0-0] = 0$$

Here, $X = \{\mathbb{E}_2,\ \mathbb{E}_3\} = \{\mathbb{E}_2,\ \mathbb{E}_{15}\} = \{\mathbb{E}_2,\ \mathbb{E}_7\}$; $Y = \mathbb{E}_3 = \mathbb{E}_{15} = \mathbb{E}_7$.
– **Case 3:** (When coalition $\mathbb{F}_r^2$ contains 2 members *i.e.* $|\mathbb{F}_r^2| = 2$ and then $\mathbb{E}_2$ is added in $\mathbb{F}_r^2$.)

$$\xi'^3_2 = 3\times\frac{2! \times (4-2-1)!}{4!} \times [v(X') - v(Y')] = 3\times\frac{2! \times (4-2-1)!}{4!} \times [4-0] = \frac{4}{4} = 1$$

$$\xi_2^3 = 3\times\frac{2! \times (4-2-1)!}{4!} \times [v(X') - v(Y')] = 3\times\frac{2! \times (4-2-1)!}{4!} \times [32-0] = \frac{32}{4}\$$$

Here, $X' = \{\mathbb{E}_2,\ \mathbb{E}_3,\ \mathbb{E}_7\} = \{\mathbb{E}_2,\ \mathbb{E}_3,\ \mathbb{E}_{15}\} = \{\mathbb{E}_2,\ \mathbb{E}_7,\ \mathbb{E}_{15}\}$; $Y' = \{\mathbb{E}_3,\ \mathbb{E}_7\} = \{\mathbb{E}_3,\ \mathbb{E}_{15}\} = \{\mathbb{E}_7,\ \mathbb{E}_{15}\}$.
– **Case 4:** (When coalition $\mathbb{F}_r^2$ contains 3 members *i.e.* $|\mathbb{F}_r^2| = 3$ and then $\mathbb{E}_2$ is added in $\mathbb{F}_r^2$.)

$$\xi'^4_2 = 1\times\frac{3! \times (4-3-1)!}{4!} \times [v(X'') - v(Y'')] = 1\times\frac{3! \times (4-3-1)!}{4!} \times [4-0] = \frac{4}{4} = 1$$

$$\xi_2^4 = 1\times\frac{3! \times (4-3-1)!}{4!} \times [v(X'') - v(Y'')] = 1\times\frac{3! \times (4-3-1)!}{4!} \times [32-0] = \frac{32}{4}\$$$

Here, $X'' = \{\mathbb{E}_2,\ \mathbb{E}_3,\ \mathbb{E}_7,\ \mathbb{E}_{15}\}$; $Y'' = \{\mathbb{E}_3,\ \mathbb{E}_7,\ \mathbb{E}_{15}\}$.

Combining Case 1, Case 2, Case 3, and Case 4; we get

$$\boldsymbol{\nabla'}_2(\mathcal{N}, v) = \xi'^1_2 + \xi'^2_2 + \xi'^3_2 + \xi'^4_2 = 0hr + 0hr + 1hr + 1hr = 2hrs$$

$$\boldsymbol{\nabla}_2(\mathcal{N}, v) = \xi_2^1 + \xi_2^2 + \xi_2^3 + \xi_2^4 = 0\$ + 0\$ + \frac{32}{4}\$ + \frac{32}{4}\$ = \frac{64}{4}\$ = 16\$$$

The value $\boldsymbol{\nabla'}_2(\mathcal{N}, v)$=2hrs indicate that the IoT device $\mathbb{E}_2$ will be allocated a chunk of sub-task with estimated time of 2hrs. On the other hand, the value $\boldsymbol{\nabla}_2(\mathcal{N}, v)$=16\$ represents the payment received by $\mathbb{E}_2$ for executing the chunk of sub-task for 2hrs.

In the grand coalition, the fraction of sub-task and the budget given to $\mathbb{E}_3$ is calculated as: (by symmetry for $\mathbb{E}_7$, and $\mathbb{E}_{15}$) is given by:

– **Case 1:** (When coalition $\mathbb{F}_r^2$ is empty *i.e.* $|\mathbb{F}_r^2| = 0$ and then $\mathbb{E}_3$ is added in $\mathbb{F}_r^2$.)

$$\xi'^1_3 = \xi^1_3 = \frac{0! \times (4-0-1)!}{4!} \times [v(\{\mathbb{E}_3\}) - v(\phi)] = \frac{0! \times (4-0-1)!}{4!} \times [0-0] = 0$$

– **Case 2:** (When coalition $\mathbb{F}_r^2$ contains 1 member *i.e.* $|\mathbb{F}_r^2| = 1$ and then $\mathbb{E}_3$ is added in $\mathbb{F}_r^2$.)

$$\xi'^2_3 = \xi^2_3 = 3 \times \frac{1! \times (4-1-1)!}{4!} \times [v(X) - v(Y)] = 3 \times \frac{1! \times (4-1-1)!}{4!} \times [0-0] = 0$$

Here, $X = \{\mathbb{E}_2,\ \mathbb{E}_3\} = \{\mathbb{E}_3,\ \mathbb{E}_{15}\} = \{\mathbb{E}_3,\ \mathbb{E}_7\}$; $Y = \mathbb{E}_2 = \mathbb{E}_{15} = \mathbb{E}_7$.
– **Case 3:** (When coalition $\mathbb{F}_r^2$ contains 2 members *i.e.* $|\mathbb{F}_r^2| = 2$ and then $\mathbb{E}_3$ is added in $\mathbb{F}_r^2$.)

$$\xi'^3_3 = 2 \times \frac{2! \times (4-2-1)!}{4!} \times [v(X') - v(Y')] = 2 \times \frac{2! \times (4-2-1)!}{4!} \times [4-0] = \frac{4}{6}$$

$$\xi^3_3 = 2 \times \frac{2! \times (4-2-1)!}{4!} \times [v(X') - v(Y')] = 2 \times \frac{2! \times (4-2-1)!}{4!} \times [32-0] = \frac{32}{6}$$

Here, $X' = \{\mathbb{E}_2,\ \mathbb{E}_3,\ \mathbb{E}_7\} = \{\mathbb{E}_2,\ \mathbb{E}_3,\ \mathbb{E}_{15}\} = \{\mathbb{E}_3,\ \mathbb{E}_7,\ \mathbb{E}_{15}\}$; $Y' = \{\mathbb{E}_2,\ \mathbb{E}_7\} = \{\mathbb{E}_2,\ \mathbb{E}_{15}\} = \{\mathbb{E}_7,\ \mathbb{E}_{15}\}$.
– **Case 4:** (When coalition $\mathbb{F}_r^2$ contains 3 members *i.e.* $|\mathbb{F}_r^2| = 3$ and then $\mathbb{E}_3$ is added in $\mathbb{F}_r^2$.)

$$\xi'^4_3 = 1 \times \frac{3! \times (4-3-1)!}{4!} \times [v(X'') - v(Y'')] = 1 \times \frac{3! \times (4-3-1)!}{4!} \times [4-4] = 0$$

$$\xi^4_3 = 1 \times \frac{3! \times (4-3-1)!}{4!} \times [v(X'') - v(Y'')] = 1 \times \frac{3! \times (4-3-1)!}{4!} \times [32-32] = 0$$

Here, $X'' = \{\mathbb{E}_2,\ \mathbb{E}_3,\ \mathbb{E}_7,\ \mathbb{E}_{15}\}$; $Y'' = \{\mathbb{E}_2,\ \mathbb{E}_7,\ \mathbb{E}_{15}\}$.

Combining Case 1, Case 2, Case 3, and Case 4; we get

$$\boldsymbol{\nabla'}_3(\mathcal{N}, v) = \xi'^1_3 + \xi'^2_3 + \xi'^3_3 + \xi'^4_3 = 0hr + 0hr + \frac{4}{6}hr + 0hr = \frac{4}{6}hr = 0.66hr$$

$$\boldsymbol{\nabla}_3(\mathcal{N}, v) = \xi^1_3 + \xi^2_3 + \xi^3_3 + \xi^4_3 = 0\$ + 0\$ + \frac{32}{6}\$ + 0\$ = \frac{32}{6}\$ = 5.33\$$$

So, we have $\boldsymbol{\nabla'}_2(\mathcal{N}, v) = 2hrs$, $\boldsymbol{\nabla}_2(\mathcal{N}, v) = 16\$$; $\boldsymbol{\nabla'}_3(\mathcal{N}, v) = 0.66hr$, $\boldsymbol{\nabla}_3(\mathcal{N}, v) = 5.33\$$; $\boldsymbol{\nabla'}_7(\mathcal{N}, v) = 0.66hr$, $\boldsymbol{\nabla}_7(\mathcal{N}, v) = 5.33\$$; and $\boldsymbol{\nabla'}_{15}(\mathcal{N}, v) = 0.66hr$, $\boldsymbol{\nabla}_{15}(\mathcal{N}, v) = 5.33\$$.

6.6 Analysis of NoTUBE

Lemma 8 *In NoTUBE, for each task $\mathbb{T}_k$ the total payment made to the IoT devices is equal to budget $\mathbb{B}_k$.*

Proof Fix a task requester $\mathbb{T}_k$ having budget $\mathbb{B}_k$. Now, each sub-task is allotted the part of total budget reserved for a task $\mathbb{T}_k$. Considering any sub-task, from the construction of NoTUBE, it can be seen that the payment made to the IoT devices is some fractional part of the budget. As, we are interested in determining that fractional value which can be determined using equation 2. When it comes to the allocation of chunks or the budget both these entities is distributed in same proportion to the IoT devices. In the more clear sense, the ratio in which the sub-task is assigned to the IoT devices in terms of the total estimated time using equation 2, in the same ratio the available budget is also distributed among them.

In this collaborative environment, as summing over the estimated time of the chunks allocated to each IoT device will lead to achieve the total estimated time of the sub-task, so as the case with the payment and budget. So, if this argument is true for any sub-task of the task then we can argue that it will be true for the remaining heterogeneous sub-tasks. Hence, summing over all the sub-tasks, the total payment made to the IoT devices associated to the task $\mathbb{T}_k$ will be equal to the allotted quota of budget $\mathbb{B}_k$. Hence, NoTUBE is budget feasible.

Lemma 9 *In our system, for any arbitrary sub-task that has the highest estimated time, the number of IoT devices assigned continuously to that sub-task so that the switch happens, is constant. If the probability of assigning IoT device to the sub-task $\mathbb{T}_j^i$ of the task $\mathbb{T}_j$ be p, then $E[X] = \frac{1}{p}$; where X is the random variable capturing the event.*

Proof The proof is presented in Appendix B.1.

Corollary 1 *The above lemma (Lemma 9) claims that NoTUBE will lead to the balanced distribution of IoT devices associated to some task $\mathbb{T}_j$ among the heterogeneous sub-tasks.*

Theorem 1 (Negative Result) *NoTUBE is non-truthful.*

Proof The proof is presented in Appendix B.2.

7 Further Enhancement of The Model

In this section, we have enhanced the model discussed in section 6 by relaxing just the constraint that each sub-task should have constant or substantial number of IoT devices associated with them. It means that, now the sub-tasks will have large number of IoT devices forming the larger coalition, so as to finish the sub-task in a collaborative fashion. By large coalition we mean that, the fair division of an itinerary among the members of the grand coalition using *Shapley value* is *intractable* or *hard*. Also, relaxing this constraint may give rise to the inclusion of those IoT devices in the coalition which are having very minimal battery power left for the execution purpose. It is to be noted that these minimal battery powered IoT devices were ignored in our previous set-up presented in section 6. As for this set-up the *Shapley value* calculation for the members of the coalitions while preserving several useful properties is NP-hard. We propose a *non-truthful budget feasible mechanism* for the enhanced version of the *task assignment problem* called *non-truthful budget feasible mechanism* for *enhanced task assignment problem* (NoTUBE-ETAP).

7.1 Proposed Mechanism: NoTUBE-ETAP

This section explains the detailing of the NoTUBE-ETAP presented in Algorithm 6.

Algorithm 6: NoTUBE-ETAP ($\mathbb{T}$, $\mathbb{B}$, $\tilde{t}$, t, τ)

Output: $\mathcal{T} \leftarrow \phi$, $\mathbb{A} \leftarrow \phi$, $\boldsymbol{P} \leftarrow \phi$
foreach $i \in \tau$ **do**
    foreach $\mathbb{T}_j \in i$ **do**
        Follow line 3-4 of Algorithm 5
        while $\mathbb{L} \neq \phi$ **do**
            Follow line 6-12 of Algorithm 5
        end
        foreach $\mathbb{T}_j^i \in \mathbb{T}_j$ **do**
            foreach $\mathbb{E}_\ell \in \mathcal{N}_j^i$ **do**

$$\mathcal{T}_\ell^j \leftarrow t_j^i \cdot \left(\tilde{t}_l / \sum_{j=1}^{\lfloor |\pi^j| / \hat{k}_j \rfloor} \tilde{t}_j \right)$$

$$\boldsymbol{P}_\ell^j \leftarrow \mathbb{B}_j^i \cdot \left(\tilde{t}_l / \sum_{j=1}^{\lfloor |\pi^j| / \hat{k}_j \rfloor} \tilde{t}_j \right)$$ // $\mathbb{B}_j^i$ is the budget associated with subtask $\mathbb{T}_j^i$.

            end
            $\mathcal{T}_j \leftarrow \mathcal{T}^j \cup \mathcal{T}_\ell^j$
            $\mathbb{A}_j \leftarrow \pi^j$
            $\boldsymbol{P}_j \leftarrow \boldsymbol{P}_j \cup \boldsymbol{P}_\ell^j$
        end
        $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_j$
        $\mathbb{A} \leftarrow \mathbb{A} \cup \mathbb{A}_j$
        $\boldsymbol{P} \leftarrow \boldsymbol{P} \cup \boldsymbol{P}_j$
    end
end
return $\mathcal{T}$, $\mathbb{A}$, $\boldsymbol{P}$

For each iteration of *for* loop in line 1-20, a time slot is considered and in each time slot the available tasks are taken care by line 2-19. The stopping condition in the *while* loop in line 4 ensures that the loop terminates only when all the IoT devices are allocated to the sub-task. Once the coalition for each sub-task is determined, next objective is to have a fair division of each sub-task as chunks and the budget as payment among the members of the coalition. Utilizing line 7-15 the required objective is achieved. In line 16, $\mathcal{T}$ keeps track of amount of time IoT devices in $\mathbb{A}$ executes the allocated task. The allocation vector and payment vector of all the winning IoT devices is held in $\mathbb{A}$ and $\boldsymbol{P}$ respectively as depicted in line 17 and 18 respectively. Finally, line 21 returns the required outputs.

Theorem 2 (Negative Result) *NoTUBE-ETAP is non-truthful.*

Proof The proof follow the similar argument provided in Theorem 1.

8 Experimental Findings

In this section, we measure the efficacy of our proposed mechanism called TUBE-TAP via simulation. TUBE-TAP is compared with the carefully crafted benchmark mechanism that is *non-truthful* in nature. The manipulative behaviour of the IoT devices in case of benchmark mechanism can be seen evidently in the simulation results. It is to be noted that, our benchmark mechanism differs in terms of *allocation* and *payment* policies from the TUBE-TAP.

In the benchmark mechanism, for each task $\mathbb{T}_j$, firstly the quality IoT devices are sorted in increasing order of their bid value. Next, in each iteration an IoT device i is picked up and a check is made whether $b_{i+1}^j + \epsilon$ is less than or equal to the remaining budget associated with the task $\mathbb{T}_j$ or not, where b_{i+1}^j is the bid value of the $(i+1)^{th}$ IoT device followed by i in the sorted ordering. If the stopping condition for the i^{th} IoT device is satisfied, then it will be declared as winner and its payment is given as $\boldsymbol{P}_i^j = b_{i+1}^j + \epsilon$. Before moving to the next iteration the paid amount is deducted from the available budget. The process terminates once the stopping condition is dissatisfied.

It is to be noted that, the ϵ value is taken as 10 for the simulation purpose. The unit of bid value and the budget is taken as \$. The experiments are carried out using Python.

8.1 Simulation Set-up

Table 1 shows the data set utilized for the simulation purpose. In our case, the experiment runs for 50 rounds and the required values are plotted by taking average over these 50 rounds. Other than this, in order to strengthen our claim, we have simulated the mechanisms for two different probability distributions independently; namely, *uniform distribution* (UD) and *normal distribution* (ND). Throughout the experiment, the bid value range (in case of

UD) for IoT devices and the budget range for the tasks are kept fixed. It is to be noted that, the budget is uniformly distributed within the given range for both ND and UD cases. Considering the case of ND, for generating the bid values of the IoT devices the *mean* is taken as 110 and *standard deviation* is taken as 15.

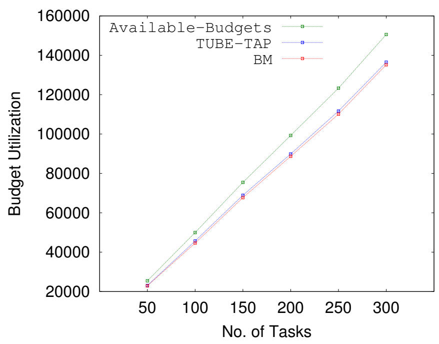
Table 1: Data set utilized for simulation purpose

Task requesters	50	100	150	200	250	300
Task executers	500	1000	1500	2000	2500	3000
Bid value range (for UD)	[80, 150]	[80, 150]	[80, 150]	[80, 150]	[80, 150]	[80, 150]
Budget distribution	[400, 600]	[400, 600]	[400, 600]	[400, 600]	[400, 600]	[400, 600]

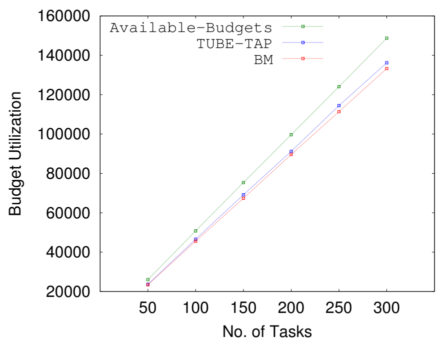
In order to measure the efficacy of TUBE-TAP, we have taken two performance metrics: 1) Budget utilization, and 2) Utility of the IoT devices.

8.2 Result Analysis

In this section, we are simulating TUBE-TAP which we are claiming is *budget feasible* and *truthful* in our setting against the benchmark mechanism (which will be referred as BM in the figures of simulation results)[4]. We can see in Fig. 5a, and Fig. 5b that the budget utilization in case of TUBE-TAP is a bit more as compared to the budget utilization in case of BM for both ND and UD cases. This is due to the fact that, in TUBE-TAP each winner is paid a value between the bid value of last winner and the bid value of the first loser present in the sorted ordering. However, in BM each winner is paid a bit more than the bid value of preceding IoT device in the ordering.



(a) Budget utilization (ND)

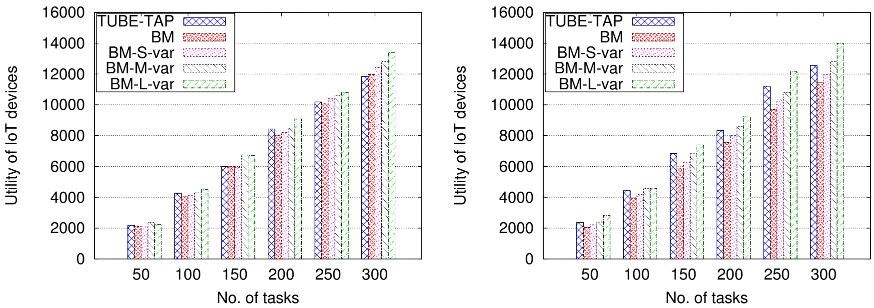


(b) Budget utilization (UD)

Fig. 5: Comparison of Budget utilization in ND and UD cases

[4] Similarly, the other two mechanisms; NoTUBE and TUBE-ETAP could be simulated and analyzed.

As the bid values of the IoT devices are sorted in increasing order, so the payment made to each winning IoT device in case of TUBE-TAP is more as compared to BM. Another important observation one can make from Fig. 5a, and Fig. 5b is that, both the mechanisms *i.e.* TUBE-TAP and BM are *budget feasible* that supports the claim made for TUBE-TAP in Lemma 2. Next comes the discussion on the behaviour of the mechanisms based on our second parameter. The sole purpose of considering this parameter is to judge the two mechanisms on the ground of *truthfulness*. During the simulation, in order to show the so called manipulative behaviour of BM we have varied the bid values of the subset of the IoT devices. More formally, we have considered that 15% of the available IoT devices (in our case this is referred as *small* variation) are increasing their bid value by 35% of their true valuation. Similar is the case with *medium* variation (30% of the available IoT devices are increasing their bid value by 35% of their true valuation) and the *large* variation (40% of the available IoT devices are increasing their bid value by 35% of their true valuation). In the figures of simulation results, BM with *small* variation, BM with *medium* variation, and BM with *large* variation is shown as BM-S-var, BM-M-var, and BM-L-var respectively. From Fig. 6a and Fig. 6b it can be seen that, most of the time the utility of IoT devices for TUBE-TAP is more as compared to the utility of IoT devices for BM in both ND and UD case.



(a) Utility of IoT devices (ND) (b) Utility of IoT devices (UD)

Fig. 6: Comparison of Utility of IoT devices in ND and UD cases

This very nature of TUBE-TAP is due to the reason that IoT devices are paid higher in case of TUBE-TAP as compared to BM that can be concluded from the results shown Fig. 5a, and Fig. 5b. Also, talking about the manipulative nature of the BM, it can be easily seen in Fig. 6a, and Fig. 6b that overall utility of the IoT devices gets increased by misreporting the bid values. The utility of IoT devices is higher in case of large variation than in case of medium variation than in case of small variation. Also, in some manipulative cases (mostly in *large* variation) it could be seen that the utility achieved by the IoT devices in case of BM bypass even the utility gained by the IoT devices in case of TUBE-TAP. So, one can conclude that larger the number of IoT devices increasing their bid value by some amount (say 35%) higher will be the utility for the IoT devices. As the IoT devices are gaining by misreporting, so BM is *non-truthful*.

9 Conclusion and Future Works

In this paper, we have investigated a heterogeneous task assignment problem in IoT based crowdsourcing through the lens of mechanism design. We have designed a *truthful* mechanism for the problem such that for each task the total payment made to the subset of IoT devices are within budget while achieving a threshold quality. Furthermore, we have considered the more realistic version of the problem by injecting the constraint that the tasks endowed with the task requesters are divisible in nature along with the several other additional constraints. We have proposed the *non-truthful budget feasible mechanisms* for the more realistic versions of the problem.

In our future works, we can think of designing a *truthful budget feasible mechanism* for the more realistic version of the problem. In addition to this, we can also think of designing a *truthful* mechanism for the set-up with multiple task requesters and multiple IoT devices, where each task requester is carrying multiple heterogeneous tasks, instead of, a single task.

Acknowledgements We would like to thanks the research students and faculty members of the Department of CSE, NIT Durgapur for their valuable suggestions during the course of this work. Further, we would like to thank Prof. Y. Narahari and members of the Game Theory Lab. at IISc Bangalore for their suggestions and directions. Finally, we thank the Government of India Ministry of Human Resource Development for their Institute scholarship fund given to the Ph.D. scholars.

References

Alfaro LD, Shavlovsky M, Polychronopoulos V (2016) Incentives for truthful peer grading. CoRR abs/1604.03178, URL http://arxiv.org/abs/1604.03178, 1604.03178

Assadi S, Hsu J, Jabbari S (2015) Online assignment of heterogeneous tasks in crowdsourcing markets. CoRR abs/1508.03593, URL http://arxiv.org/abs/1508.03593, 1508.03593

Bhat S, Padmanabhan D, Jain S, Narahari Y (2016) A truthful mechanism with biparameter learning for online crowdsourcing: (extended abstract). In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, AAMAS '16, pp 1385–1386, URL http://dl.acm.org/citation.cfm?id=2937029.2937172

Chatzimilioudis G, Konstantinidis A, Laoudias C, Zeinalipour-Yazti D (2012) Crowdsourcing with smartphones. IEEE Internet Computing 16(5):36–44, DOI 10.1109/MIC.2012.70

Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) Introduction to algorithms. MIT press

Daniel F, Kucherbaev P, Cappiello C, Benatallah B, Allahbakhsh M (2018) Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. ACM Computing Surveys 51(1):7:1–7:40, DOI 10.1145/3148148, URL http://doi.acm.org/10.1145/3148148

Duan Z, Tian L, Yan M, Cai Z, Han Q, Yin G (2017) Practical incentive mechanisms for iot-based mobile crowdsensing systems. IEEE Access 5:20383–20392, DOI 10.1109/ACCESS.2017.2751304

Fatima SS, Wooldridge M, Jennings NR (2008) A linear approximation method for the shapley value. Artificial Intelligence 172(14):1673–1699, DOI https://doi.org/10.1016/j.artint.2008.05.003

Feng Z, Zhu Y, Zhang Q, Ni LM, Vasilakos AV (2014) TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, Toronto, ON, Canada, pp 1231–1239, DOI 10.1109/INFOCOM.2014.6848055

Gao L, Hou F, Huang J (2015a) Providing long-term participation incentive in participatory sensing. In: 2015 IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, pp 2803–2811, DOI 10.1109/INFOCOM.2015.7218673

Gao Y, Chen Y, Liu KJR (2015b) On cost-effective incentive mechanisms in microtask crowdsourcing. IEEE Transactions on Computational Intelligence and AI in Games 7(1):3–15, DOI 10.1109/TCIAIG.2014.2298361

Goel G, Nikzad A, Singla A (2014) Mechanism design for crowdsourcing markets with heterogeneous tasks. In: Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2014, November 2-4, 2014, Pittsburgh, Pennsylvania, USA

Gong X, Shroff N (2018) Incentivizing truthful data quality for quality-aware mobile data crowdsourcing. In: Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, ACM, New York, NY, USA, Mobihoc'18, pp 161–170, DOI 10.1145/3209582.3209599, URL http://doi.acm.org/10.1145/3209582.3209599

Howe J (2006) The rise of crowdsourcing. Wired Magazine 14(6), URL http://www.wired.com/wired/archive/14.06/crowds.html

Jain S, Ghalme G, Bhat S, Gujar S, Narahari Y (2016) A deterministic mab mechanism for crowdsourcing with logarithmic regret and immediate payments. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, AAMAS '16, pp 86–94, URL http://dl.acm.org/citation.cfm?id=2936924.2936941

Jain S, Gujar S, Bhat, Zoeter O, Narahari Y (2018) A quality assuring, cost optimal multi-armed bandit mechanism for expertsourcing. Artificial Intelligence 254:44 – 63, DOI https://doi.org/10.1016/j.artint.2017.10.001, URL http://www.sciencedirect.com/science/article/pii/S000437021730125X

Kobayashi M, Morita H, Matsubara M, Shimizu N, Morishima A (2018) An empirical study on short- and long-term effects of self-correction in crowdsourced microtasks. In: HCOMP, AAAI Press, pp 79–87

Lee JS, Hoh B (2010a) Dynamic pricing incentive for participatory sensing. Elsevier Journal of Pervasive and Mobile Computing 6(6):693–708, DOI https://doi.org/10.1016/j.pmcj.2010.08.006

Lee JS, Hoh B (2010b) Sell your experiences: a market mechanism based incentive for participatory sensing. In: 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom), Mannheim, Germany, pp 60–68, DOI 10.1109/PERCOM.2010.5466993

Li Y, Zhao Y, Ishak S, Song H, Wang N, Yao N (2018) An anonymous data reporting strategy with ensuring incentives for mobile crowd-sensing. Journal of Ambient Intelligence and Humanized Computing 9(6):2093–2107, DOI 10.1007/s12652-017-0529-x, URL https://doi.org/10.1007/s12652-017-0529-x

Luo T, Das SK, Tan HP, Xia L (2016) Incentive mechanism design for crowdsourcing: An all-pay auction approach. ACM Transactions on Intelligent Systems and Technology 7(3):35:1–35:26, DOI 10.1145/2837029, URL http://doi.acm.org/10.1145/2837029

Maschler M, Solan E, Zamir S (2013) The shapley value. In: Game Theory, Cambridge University Press, pp 748–781, DOI 10.1017/CBO9780511794216.019

Mazlan N, Syed Ahmad SS, Kamalrudin M (2018) Volunteer selection based on crowdsourcing approach. Journal of Ambient Intelligence and Humanized Computing 9(3):743–753, DOI 10.1007/s12652-017-0490-8, URL https://doi.org/10.1007/s12652-017-0490-8

Nisan N, Roughgarden T, Tardos E, Vazirani VV (2007) Algorithmic Game Theory. Cambridge University Press, New York, NY, USA

Reddy S, Estrin D, Hansen M, Srivastava M (2010) Examining micro-payments for participatory sensing data collections. In: 12^{th} International Conference on Ubiquitous Computing (UbiComp), ACM, Copenhagen, Denmark, pp 33–36, DOI 10.1145/1864349.1864355

Roughgarden T (2016) CS269I: Incentives in computer science (Stanford University course). Lecture #17: Scoring Rules and Peer Prediction (Incentivizing Honest Forecasts and Feedback)

Shapley LS (1953) A value for n-person games. In: Kuhn HW, Tucker AW (eds) Contributions to the Theory of Games II, Princeton University Press, Princeton, pp 307–317

Singer Y (2010) Budget feasible mechanisms. In: Proceedings of the 2010 IEEE 51^{st} Annual Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, USA, FOCS '10, pp 765–774, DOI 10.1109/FOCS.2010.78

Singh VK, Mukhopadhyay S, Xhafa F (2018a) A budget feasible peer graded mechanism for IoT-based crowdsourcing. CoRR abs/1809.09315, URL http://arxiv.org/abs/1809.09315, 1809.09315

Singh VK, Mukhopadhyay S, Xhafa F, Sharma A (2018b) A budget feasible mechanism for hiring doctors in e-healthcare. In: 2018 IEEE 32^{nd} International Conference on Advanced Information Networking and Applications (AINA), Krakow, Poland, pp 785–792, DOI 10.1109/AINA.2018.00117

Slivkins A, Vaughan JW (2014) Online decision making in crowdsourcing markets: Theoretical challenges. SIGecom Exchanges 12(2):4–23, DOI 10.1145/2692359.2692364, URL http://doi.acm.org/10.1145/2692359.2692364

Tinati R, Madaan A, Hall W (2017) The role of crowdsourcing in the emerging internet-of-things. In: Proceedings of the 26th International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, WWW '17 Companion, pp 1669–1672, DOI 10.1145/3041021.3051693, URL https://doi.org/10.1145/3041021.3051693

Xu P, Srinivasan A, Sarpatwar KK, Wu K (2017) Budgeted online assignment in crowdsourcing markets: Theory and practice. In: Proceedings of the 16^{th} Conference on Autonomous Agents and Multi-Agent Systems, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, AAMAS'17, pp 1763–1765, URL http://dl.acm.org/citation.cfm?id=3091125.3091431

Yuen M, King I, Leung K (2011) A survey of crowdsourcing systems. In: 3^{rd} IEEE International Conference on Privacy, Security, Risk and Trust and 3^{rd} IEEE International Conference on Social Computing, Boston, MA, USA, pp 766–773, DOI 10.1109/PASSAT/SocialCom.2011.203

Zhao D, Li X, Huadong M (2014) How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In: Proceeding of Annual IEEE International Conference on Computer Communications, ACM, Toronto, Canada, pp 173–184, DOI 10.1109/INFOCOM.2014.6848053

Appendix

A Omitted Proofs from Section 5

A.1 Proof of Lemma 2

Fix a task requester $\mathbb{R}_j$ and a task $\mathbb{T}_j$. From the construction of TUBE-TAP, it is clear that, the maximum payment that any winning IoT device will be paid is $\frac{\mathbb{B}_j}{k}$; where k is the largest index obtained in the ordering of IoT devices that satisfies $b_k^j \leq \frac{\mathbb{B}_j}{k}$. Now, the total payment $\boldsymbol{P}_j$ is given as:

$$\boldsymbol{P}_j = \sum_{\mathbb{E}_i \in \mathbb{A}_j} \boldsymbol{P}_i^j \leq \sum_{\mathbb{E}_i \in \mathbb{A}_j} \frac{\mathbb{B}_j}{k} = \frac{\mathbb{B}_j}{k} \times k = \mathbb{B}_j$$

From here we can say that, $\boldsymbol{P}_j \leq \mathbb{B}_j$. As this is true for any task $\mathbb{T}_j$, so the budget feasibility will hold for all the available tasks *i.e.* $\sum\limits_{\mathbb{A}_j \in \mathbb{A}} \sum\limits_{\mathbb{E}_i \in \mathbb{A}_j} \boldsymbol{P}_i^j \leq \sum\limits_{\mathbb{T}_j \in \mathbb{T}} \mathbb{B}_j$. This completes the proof.

A.2 Proof of Lemma 4

By the definition of indicator random variable, we can write X_j^i is 1 when $\mathbb{U}$ occurs and 0 when $\mathbb{U}$ does not occurs. So, as $X_j^i = \mathrm{I}\{\mathbb{U}\}$. Taking expectation both side, we get from definition of expectation

$$E[X_j^i] = E[I\{\mathbb{U}\}] = Pr\{\mathbb{U}\}$$

The detailing of this lemma is provided in draft version Singh et al. (2018a).

A.3 Proof of Lemma 6

Fix an IoT device $\mathbb{E}_i$. In similar line the proof is illustrated in Cormen et al. (2009). Our proof is divided into two cases. From Lemma 5 it can be seen that the probability that $\mathbb{E}_i$ will be considered for any task $\mathbb{T}_j$ is p. Let $X_{kl}^i = I\{A_{kl}^i\}$ be the indicator random variable associated with an event that the IoT device $\mathbb{E}_i$ is rejected for at least l tasks starting form k^{th} task. It is to be noted that, the participation in one time slot by the IoT device is independent of the participation in other time slots. So, for any given event X_{kl}^i, the probability that for all l tasks the IoT device is rejected is given as

$$Pr\{A_{kl}^i\} = p \cdot p \ \cdots l \ times = p^l \tag{9}$$

As in our case, k varies from 1 to $k_i - l + 1$ (i.e. $1 \leq k \leq k_i - l + 1$), so the total number of such rejections could be formulated as:

$$Y = \sum_{k=1}^{k_i - l + 1} X_{kl}^i$$

Taking expectation both side, we get

$$E[Y] = E\left[\sum_{k=1}^{k_i - l + 1} X_{kl}^i\right]$$

By linearity of expectation, we get

$$= \sum_{k=1}^{k_i - l + 1} E[X_{kl}^i]$$

From the definition of expectation in Lemma 4, we have

$$E[Y] = \sum_{k=1}^{k_i-l+1} Pr\{A^i_{kl}\}$$

Using equation 9, we get

$$= \sum_{k=1}^{k_i-l+1} p^l$$

$$E[Y] = (k_i - l + 1) \cdot p^l$$

Now, for $l = c \log_p k_i$ and for some positive constant c, we obtain

$$\begin{aligned} E[Y] &= (k_i - c\log_p k_i + 1) \cdot p^{c \log_p k_i} \\ &= (k_i - c\log_p k_i + 1) \cdot k_i^c \\ &= k_i^{c+1} - ck_i^c \log_p k_i + k_i^c \\ &= \Theta(k_i^c) \end{aligned}$$

From here we can conclude that, for some constant $c \geq 1$ the longest continuous rejection boils down to $\Theta(\log_p k_i)$. Hence, the claim survived.

A.4 Proof of Lemma 7

Fix an IoT device $\mathbb{E}_i$. As $\mathbb{E}_i$ has shown interest on k_i tasks that are present in different time slots. The probability that $\mathbb{E}_i$ will be considered for task $\mathbb{T}_j$ is p ($\Pr\{\mathbb{E}_i$ is not considered for task $\mathbb{T}_j\} = 1 - p$). Also, it can be seen that, the consideration of $\mathbb{E}_i$ in any time slot is independent of other time slots. So, the probability that $\mathbb{E}_i$ will not be considered at all for any of the k_i tasks is given as:

$$\begin{aligned} Pr[X^i < 1] &= (1-p) \cdot (1-p) \ldots k_i \ times \\ &= (1-p)^{k_i} \end{aligned}$$

Following the inequality $1 + x \leq e^x$, we get

$$Pr[X^i < 1] \leq e^{-p \cdot k_i} = \frac{1}{e^{p \cdot k_i}}$$

Now, the probability that any $\mathbb{E}_i$ will be considered at least once is given as

$$Pr[X^i \geq 1] \geq \left(1 - \frac{1}{e^{p \cdot k_i}}\right)$$

Hence, the claim survives. Also, for $p = \ln 2$, we can see that

$$Pr[X^i \geq 1] \geq \left(1 - \frac{1}{e^{\ln 2 \cdot k_i}}\right)$$

$$= \left(1 - \frac{1}{2k_i}\right)$$

It can be concluded that, the term $\frac{1}{2k_i}$ represents that any arbitrary $\mathbb{E}_i$ will not be considered at all is very small, and can say that it is very unlikely to occur. So, the term $(1 - \frac{1}{2k_i})$ will be quite large and hence can say that any IoT device could be considered for at least once with larger probability.

B Omitted Proofs from Subsection 6.6

B.1 Proof of Lemma 9

At any i^{th} iteration of the Algorithm 5, say a sub-task $\mathbb{T}_j^i$ of task $\mathbb{T}_j$ is having the highest estimated time among the other sub-tasks. From the construction of NoTUBE, the sub-task $\mathbb{T}_j^i$ is a potential candidate for getting the IoT device(s). It is quite clear from Algorithm 5 that, with each assignment of IoT device to the sub-task $\mathbb{T}_j^i$, an estimated time of the sub-task under consideration goes down monotonically. Once the substantial number of IoT devices is allocated to sub-task $\mathbb{T}_i^j$, a new sub-task comes up as a potential candidate for getting the IoT devices, we call this as *switch*. So, in this lemma we are trying to answer the query: *what is the substantial number of IoT devices in expectation that is to be assigned to any sub-task with highest estimated time before the switch happens?*

Let us suppose the assignment of substantial number of IoT devices to a sub-task as a *Bernoulli trials*. So, we have a sequence of *Bernoulli trials*, each with probability of assigning IoT device to the sub-task $\mathbb{T}_j^i$ be p and the probability of not assigning the IoT device to the sub-task $\mathbb{T}_j^i$ is $q = 1-p$. It can be observed that, the switch can occur after 1^{st} assignment, or 2^{nd} assignment, or 3^{rd} assignment and so on. Making this argument more explicit, we let X_i be the random variable associated with an event in which the switch occurs after i^{th} assignment: $X_i = \mathrm{I}\{$*switch* occurs after i^{th} assignment$\}$. Let X be the random variable denoting the total number of assignment before the switch happens. So, X has values in the range $\{1, 2, 3, \ldots\}$ and for $x \geq 1$,

$$Pr\{X = x\} = \underbrace{(1-p) \cdot (1-p) \cdot \ldots \cdot (1-p)}_{x\text{-1 times}} \cdot p \tag{10}$$

since we have $x - 1$ number of IoT device(s) assigned to any sub-task before the *switch* occurs. It can be seen that, the probability distribution in equation 10 is said to be *geometric distribution*. Now, the expected value of random variable in which we are interested in is given as:

$$E[X] = \sum_{x=1}^{\infty} x \cdot (1-p)^{x-1} \cdot p$$

$$= \frac{p}{p-1} \cdot \sum_{x=0}^{\infty} x \cdot (1-p)^{x}$$

$$
\begin{aligned}
&= \frac{p}{p-1} \cdot \frac{1-p}{p^2} \\
&= \frac{p}{p-1} \cdot \frac{1-p}{p} \cdot \frac{1}{p} \\
E[X] &= \frac{1}{p}
\end{aligned}
$$

Now, if we have value of p as $\frac{1}{2}$ then we get $E[X] = 2$, which is a small constant. Similarly, for $p = \frac{1}{3}$ we have $E[X] = 3$. Hence the claim survives.

B.2 Proof of Theorem 1

Fix a task $\mathbb{T}_j$. For any arbitrary sub-task, it may happen that the subset of the IoT devices that are the member of the grand coalition can form the sub-coalition (*coalition other than the grand coalition*) and could achieve the estimated completion time of the sub-task. The result of which, two things could be observed: (i) the IoT devices in the sub-coalition will have to utilize more of their battery power as they will be getting larger part of the sub-task this time as compared to the part they were getting in the grand coalition but within their battery power drainage time; (ii) the IoT devices in the sub-coalition is guaranteed to be paid more than they were in the grand coalition.

The benefit raised in point (ii) from IoT device point of view will motivate the IoT devices to form the sub-coalition and gain. Hence, the IoT devices are gaining by manipulating the system. So, NoTUBE is not truthful.