Mechanism Design for Crowdsourcing Markets with Heterogeneous Tasks

Gagan Goel
Google, NY
gagangoel@google.com

Afshin Nikzad
Stanford
nikzad@stanford.edu

Adish Singla
ETH Zurich
adish.singla@inf.ethz.ch

Abstract

Designing optimal pricing policies and mechanisms for allocating tasks to workers is central to online crowdsourcing markets. In this paper, we consider the following realistic setting of online crowdsourcing markets – we are given a set of heterogeneous tasks requiring certain skills; each worker has certain expertise and interests which define the set of tasks she is interested in and willing to do. Given this bipartite graph between workers and tasks, we design our mechanism TM-UNIFORM which does the allocation of tasks to workers, while ensuring budget feasibility, incentive-compatibility and achieves near-optimal utility. We further extend our results by exploiting a link with online Adwords allocation problem and present a randomized mechanism TM-RANDOMIZED with improved approximation guarantees. Apart from strong theoretical guarantees, we carry out extensive experimentation using simulations as well as on a realistic case study of Wikipedia translation project with Mechanical Turk workers. Our results demonstrate the practical applicability of our mechanisms for realistic crowdsourcing markets on the web.

Introduction

Motivated by a realistic crowdsourcing task of translating Wikipedia articles, in this paper, we study the following question:

How does one design market mechanisms for crowdsourcing when the tasks are heterogeneous and workers have different skill sets?

The recent adoption of crowdsourcing markets on Internet has brought increasing attention to the scientific questions around the design of such markets. A common theme in these markets is that there is a *requester* who has a limited budget and a set of tasks to accomplish by a pool of online workers, for instance, on platforms such as Amazon's Mechanical Turk[1] (henceforth, MTurk), ClickWorker[2], and CrowdFlower[3]. The crowdsourcing tasks are of variety of nature including image annotation, rating search engine results, validating recommendation engines, collection of labeled data, and text translation.

Incentives and Market Efficiency: A key to making these markets efficient is to design proper incentive structures and pricing policies for workers. Due to the financial constraints of the requester, pricing the tasks too high can result in lower output for the requestor. On the other hand, pricing the tasks too low can disincentivize workers to work on the tasks. This trade-off between efficiency and workers' incentives makes the pricing decisions in crowdsourcing markets complex, and thus we require new algorithms that take into account both the strategic behavior of workers and the limited budget of the requester.

Workers with different skill sets and heterogeneous tasks: In a realistic crowdsourcing setting, each worker has certain expertise and interests which define the set of tasks she can and is willing to do. For instance, consider a set of heterogeneous task of translating Wikipedia articles into different languages. Here a tuple of topic of the articles and a target language represents a unique task. Clearly, based on the worker's language skills and topic expertise, she can only translate some articles into some languages, and not all. There are numerous other crowdsourcing scenarios where the tasks require specialized knowledge to accomplish them. Mathematically speaking, this results in a bipartite graph between workers and tasks, and can thus require techniques from matching theory to achieve optimal allocation of tasks to workers.

Budget-feasible mechanisms: A series of recent results (Singer 2010; Singer 2011; Singla and Krause 2013a; Chen, Gravin, and Lu 2011; Singla and Krause 2013b) have proposed the use of budget feasible procurement auction (first introduced by Singer (2010)), as a framework to design market mechanisms for crowdsourcing. However, the current results are limiting in the sense that they make a simplifying assumption that tasks are homogeneous or they don't consider the matching constraints given by the bipartite graph as described above. Technically speaking, these simplifications help in the sense that the mechanism has to focus on picking the right set of workers only, whereas in our setting it has to do both - pick the right set of workers and assign them to the right set of tasks, while maintaining the efficiency and truthfulness properties.

[1] https://www.mturk.com/
[2] http://www.clickworker.com
[3] http://www.crowdflower.com/

Our Results

In this paper, we look at the incentive-compatible mechanism design problem for the following setting – there is a requester who has a set of heterogeneous tasks and a limited budget. For each task, there is a fixed utility that the requester achieves if that task gets completed. To do the tasks, there is a pool of workers. Each worker has certain skill sets and interests which makes her eligible to do only certain tasks, and not all. Moreover, each worker has a cost, which is the minimum amount she is willing to take for doing a task. This minimum cost is assumed to be a private information of the worker, and is same for all the tasks. For expositional simplicity, we assume that a worker can do only one task (we will relax this constraint later). The goal is to design an auction mechanism that is: i) incentive compatible in the sense that it is truthful for agents to report their true cost, ii) picks a set of workers and assigns each to a task such that the utility of the requester is maximized, and iii) budget feasible, *i.e.*, the total payments made to the workers does not exceed the budget of the requester.

We begin by designing a deterministic mechanism for the above problem which we call as TM-UNIFORM (*i.e.* Truthful Matching using Uniform Rate). We first give a mechanism that is not fully truthful but satisfies truthfulness in a weaker form, which we call oneway-truthfulness. Briefly, by this property, workers only have incentive to report costs lower than their true cost. This property also comes handy in analyzing the performance of the mechanism, showing that it achieves an approximation factor of 3, compared to the optimum solution (when the costs of the workers were known to the requester). Then, we design a new payment rule for TM-UNIFORM, that makes it fully truthful.

To improve the approximation guarantees of our mechanism, we make an interesting connection of a subroutine in TM-UNIFORM to the well studied problem of online bipartite-matching and Adwords allocation problem. We use this connection (in particular a result from Goel and Mehta (2008)) to design a randomized mechanism TM-RANDOMIZED with approximation factor of $\frac{2e-1}{e-1} \approx 2.58$. However, for this mechanism, we can only show that it is what we call *truthful in large markets*, that is, the incentive to deviate goes down to zero as the market grows larger.

Finally, we carry out extensive experimentation on a realistic case study of Wikipedia translation project using Mechanical Turk workers. Our results demonstrate the practical applicability of our mechanism. We also do simulations on synthetic data to evaluate the performance of our mechanisms on various parameters of the problem.

We note that our mechanisms easily extend to work for many-to-many matchings as well (where each task needs to be done several times and each worker can do multiple tasks), even though we describe all of our mechanisms for the simple case where each worker is willing to do at most one task and each task needs to be done by at most one worker. More interestingly, in the many-to-many setting, we can handle the case when the utility of doing a task is a non-decreasing concave function of the number of times that the task is done.

Related Work

From a technical perspective, the most similar work to that of ours is the design of budget-feasible mechanisms, initiated in Singer (2010). Subsequent research in this direction (Chen, Gravin, and Lu 2011; Bei et al. 2012; Singer 2011; Singla and Krause 2013a) has improved the current results and extended them to richer models and applications. At the heart of it, these results consider two models – one is where each worker provides a fixed utility to the requester if she gets hired (*i.e.* mechanism design version of the knapsack problem), other is when there is a general utility function (assumed to be submodular) on the set of workers that get picked. For the knapsack utility function, the best known approximation ratio is $2 + \sqrt{2}$ (for deterministic mechanisms), and 3 (for randomized mechanisms), given by Chen, Gravin, and Lu (2011). For submodular utility functions, the best known approximation ratio is 8.34 (for an exponential-time mechanism) and 7.91 (for randomized mechanisms), given by Chen, Gravin, and Lu (2011). By using the assumption of large markets, the approximation ratios are improved to 2 for knapsack functions and 4.75 for submodular functions (Singla and Krause 2013a).

We note that our model generalizes the results of budget feasible mechanism design by extending them to problems with matching constraints, though we consider a simpler utility functions (*i.e.*, we consider knapsack and non-decreasing concave utility functions, instead of the more general class of submodular functions). The problem of knapsack utility functions with matching constraints has been studied by Singer (2010) and the proposed mechanism achieves an approximation ratio of 7.3. However, we make explicit use of the mathematical structure of matchings in bipartite graphs and the assumption of large markets to design *polynomial-time* deterministic and randomized mechanisms with much better approximation ratios as compared to what is given by the current known results.

Other related work in this area studies the budget-feasible mechanism design problem in an online learning setting. Some relevant results in this direction are (Badanidiyuru, Kleinberg, and Singer 2012; Singla and Krause 2013b; Singer and Mittal 2013). Motivated from crowdsourcing settings, budget-limited multi-armed bandits have also been studied (Badanidiyuru, Kleinberg, and Slivkins 2013; Tran-Thanh et al. 2010; Tran-Thanh et al. 2012a; Tran-Thanh et al. 2012b). The issue of heterogenous tasks and workers having skill sets that restricts the set of tasks they can do was studied from an online algorithm design perspective by Ho and Vaughan (2012). Another recent work (Difallah, Demartini, and Cudré-Mauroux 2013) focuses on automated tools to pick the right set of eligible workers for a given task based on the profile of the workers. There has also been some work on understanding the issue of workers' incentives in crowdsourcing markets more closely. A model of workers is proposed in Horton and Chilton (2010) in order to estimate their wages, and Horton and Zeckhauser (2010) presents an automated way to negotiate payments with workers.

We would like to point that some of the recent advancements in the theory of online algorithms for matching and allocation problems (Karp, Vazirani, and Vazirani 1990; Goel and Mehta 2008; Aggarwal et al. 2011; Devanur and Hayes 2009; Devanur, Jain, and Kleinberg 2013) inspired from online advertising are also relevant for the crowdsourcing setting. In fact, we use one of the technical result of Goel and Mehta (2008) in our randomized mechanism to improve the approximation ratio.

The Model

We model the market with a bipartite Graph $G(P, T)$ where P is the set of people (workers) and T is the set of tasks. For any person $p \in P$, let c_p denote its cost, which is assumed to be private information of person p. Also, let u_t denote the utility of a task $t \in T$. An edge $e = (p, t)$ in the graph indicates that person p can do task t. Also we denote the budget of the requester by B. We make a *large market assumption* which is formally defined below.

A matching in G is an assignment of tasks to people such that each task is assigned to at most one person and each person is assigned at most one task. The goal is to design a mechanism that solicits bids from people (representing their private costs), and outputs a matching[4] which represents the recruited people and the tasks that are allocated to them. In addition, the mechanism comes up with a payment for each recruited person. The properties that the mechanism has to satisfy are: i) *Truthfulness*, that is, reporting the true cost should be the dominant strategy of the people, and ii) *Budget-feasibility*, that is, the total payment shouldn't exceed the budget B. The mechanism has to achieve these two properties while trying to maximize the total utility obtained from the tasks that get allocated.

Large Markets

Crowd-sourcing systems are excellent examples of *large markets*. Informally speaking, a market is said to be large if the number of participants are large enough that no single person can affect the market outcome significantly. Our results take advantage of this nature of the crowdsourcing markets to design better mechanisms. Formally speaking, we assume that in our market, the utility of a single task is very small compared to the overall utility of the optimal solution. In other words, the ratio $\theta = \frac{u_{\mathsf{max}}}{U^*}$ is small, where $u_{\mathsf{max}} = \max_{t \in T} u_t$ and U^* is the maximum utility that can be gained by assignment of tasks to people which is budget feasible.

It is worth pointing out that such assumptions have been considered before in large-market matching problems; for a well-known example, we refer to Mehta et al. (2007) where the *small bid to budget ratio* assumption is considered, *i.e.* they assume that the ratio of the maximum bid in the market is (arbitrarily) small compared to the budget.

Definitions

We now introduce some notation and useful definitions in order to describe our mechanisms. Let $N(p)$ and $N(t)$ respectively denote the set of neighbors of a person p and a task t in the graph G. Also, for simplicity, we sometimes denote $E(G)$ by E.

For a matching M and a person $p \in P$, the match of p in M is denoted by $\mathsf{M}(p)$ (possibly equal to $\emptyset$). *Cost* of M, denoted by $c(\mathsf{M})$, is defined as $\sum_{(p,t)\in \mathsf{M}} c_p$. Also, *utility* of M, denoted by $u(\mathsf{M})$ is defined as $\sum_{(p,t)\in \mathsf{M}} u_t$.

For any two matchings M, N, let $\mathsf{M} \triangle \mathsf{N}$ denote the graph which contains only the edges that appear in exactly one of the matchings M, N. It is a well-known fact that such a graph is always a union of vertex-disjoint paths and cycles.

We compare the performance of our mechanism to the optimum solution that knows the people's costs (denoted by offline optimum). We say that a mechanism has an approximation ratio of α (where $\alpha \geq 1$) if the utility obtained by this mechanism is always at least $\frac{1}{\alpha}$ of the utility obtained by the offline optimum solution.

The Uniform Mechanism (TM-UNIFORM)

In this section, we present a simple mechanism TM-UNIFORM (*i.e.* Truthful Matching using Uniform Rate). The mechanism pays the workers in a uniform manner, *i.e.* if a worker is assigned a task with utility u, then it will be paid $r \cdot u$, where coefficient r is the same for all workers. The coefficient r is called the *buck per bang* rate of the mechanism; it will be discussed in more details below.

The mechanism, although not being truthful, satisfies truthfulness in a weaker form, which we call *oneway-truthfulness*, formally defined below. Briefly, by this property, if a player has incentive to report untruthfully, then she only has incentive to report a cost lower than her true cost. This property also comes handy in analyzing the performance ratio of the mechanism, showing that it achieves an approximation factor of 3, compared to the optimum solution (*i.e.*, the maximum achievable utility when the true costs of the workers are known to the requester). We make the same mechanism truthful by changing its payment rule and designing a non-uniform payment rule.

Oneway-truthfulness

Consider a reverse auction in which there exists a set of sellers P where each seller $p \in P$ has a private cost c_p. In a truthful mechanism, no seller wants to report a fake cost regardless of what others do. In a oneway-truthful mechanism, no seller wants to report a cost higher than its true cost regardless of what others do. This notion is formally defined below. For clarification, we first define the notion of cost vector briefly: when we say a cost vector d, we mean a vector which has an entry d_p corresponding to any player p, where d_p represents the cost associated with player p.

Definition 1. *A mechanism $\mathcal{M}$ is oneway-truthful if for any seller $p \in P$ and any cost vector d for which $d_p > c_p$ we have*

$$u_p(c_p, d_{-p}) \geq u_p(d_p, d_{-p})$$

where d_{-p} denotes the cost vector corresponding to the rest of sellers except p and $u_p(x, d_{-p})$ denotes the utility of p when she reports x and other players report d_{-p}.

[4] we relax this constraint later to include the case when a person can do multiple tasks and each task can be done multiple times

Description of the Mechanism

The key concept in the mechanism is a *buck per bang* rate r (or simply the *rate*) representing the payment that the mechanism is willing to pay per unit of utility, *i.e.* if a worker is assigned a task with utility u, then it will be paid $r \cdot u$. Here, the coefficient r is same for all the workers. The buck per bang rate of an edge $e = (p, t)$, denoted by $\mathsf{bb}(e)$, is defined by $\frac{c_p}{u_t}$. Also, let $G(r)$ be a subgraph of G which only contains edges with rate at most r. The mechanism uses a fixed (and arbitrary) permutation of the vertices of P, which we denote by the permutation σ.

TM-UNIFORM starts with $r = \infty$ and it gradually decreases the rate r. Let $m = |E(G)|$ and $e_1, \ldots, e_m$ be a list in which the edges are sorted w.r.t. their buck per bang rate in decreasing order, *i.e.* for e_i and e_j, we have $i \leq j$ iff $\mathsf{bb}(e_i) \geq \mathsf{bb}(e_j)$. Also, for technical reasons, let e_0 be an isolated dummy edge with buck per bang rate of infinity.

The mechanism is formally presented in Procedure TM-Uniform. For any fixed r, it constructs the graph $G' = G(r)$ and calls Procedure FindMatching to find a matching or assignment $\mathsf{M} \subseteq E(G')$ in G'. Procedure FindMatching takes as input a fixed permutation σ of the nodes in P. Then, the nodes in P are visited one by one in the order of appearance in σ. When p is visited, the mechanism assigns p to a task t which has the highest utility among all the tasks that can be currently assigned to p. Let M denote the matching returned by the procedure after visiting all the nodes in P. If $r \cdot u(\mathsf{M}) > B$, then the mechanism decreases the rate r slightly and repeats this procedure for the new r; otherwise, it stops.

To give more intuition on what TM-UNIFORM does, we can think of the rate r as a line that sweeps the sorted list $e_1, \ldots, e_m$ from left to right in a continuous motion. All the edges that have buck per bang rate more than r fall to the left of the line. In case of ties (in buck per bang rates), the edges fall to the left of the line *one by one* in the order of their appearance in the list. The graph G' always contains all the edges to the right of the line. It stops when the matching produced by FindMatching(G', σ) has utility at most B/r. Let r^*, G^* respectively denote r, G' when mechanism stops. Based on this description, we define a notion of *time* for the mechanism, which will be used in the analysis.

Definition 2. *During execution of the Mechanism TM-Uniform, we say that the mechanism is at* iteration (r, e) *if the last edge that has been removed from* G' *is* e *and the current rate (position of the sweep line) is* r.

We emphasize that according to the above definition, we have a continuum of iterations each of which correspond to a value of r as it is decreasing continuously (when the line sweeps the sorted list). To be more precise, there can also be two different iterations (r, e) and (r, e') corresponding to the same value of r, which happens when $r = \mathsf{bb}(e) = \mathsf{bb}(e')$.

The mechanism uses a uniform payment scheme, *i.e.* paying each worker $r \cdot u_{\mathsf{M}(p)}$ (where $\mathsf{M}(p)$ denotes the task assigned to p, possibly equal to $\emptyset$). With this payment, mechanism TM-UNIFORM satisfies truthfulness in a weaker form, which we call *oneway-truthfulness* (*i.e.* players only have incentive to report costs lower than their true cost). This uniform payment scheme makes it easy to analyze the performance of the mechanism.

Procedure FindMatching

input : Graph $G'(P, T)$, Permutation σ
output: A matching in G'

$\mathsf{M} \leftarrow \emptyset$;
$T' \leftarrow T$;
for $i \leftarrow 1$ **to** $|P|$ **do**
  Find the task t with the highest utility such that $t \in N(\sigma(i)) \cap T'$;
  $\mathsf{M} \leftarrow \mathsf{M} \cup (\sigma(i), t)$;
  $T' \leftarrow T' \backslash \{t\}$;
Return the matching M;

Procedure TM-Uniform

input : Graph $G(P, T)$, Budget B, Permutation σ
output: A matching in G

$G' = G$;
for $i \leftarrow 1$ **to** m **do**
  $\mathsf{M} = $ FindMatching(G', σ);
  if $\mathsf{bb}(e_i) \cdot u(\mathsf{M}) \leq B$ **then**
    $r \leftarrow \min\left(\frac{B}{u(\mathsf{M})}, \mathsf{bb}(e_{i-1})\right)$;
    break;
  $E(G') \leftarrow E(G') - \{e_i\}$;
Return M as the final matching;
Make the uniform payments with rate r.

Next, we state our results for TM-UNIFORM based on the uniform payment rule.

Theorem 1. TM-UNIFORM*, based on uniform rate payment rule, is budget feasible, individually rational, oneway-truthful, and is* 3*-approximate compared to the optimum solution (which assumes access to the true costs).*

The Truthful Mechanism

We can modify TM-UNIFORM and make it fully truthful by modifying the payment rule. The allocation rule (selecting the matching) stays identical to Mechanism TM-Uniform. This modified payment rule along with the allocation rule in Mechanism TM-Uniform, gives a truthful mechanism. The payment rule is in fact the so-called threshold payment rule.

The Non-uniform Payment Rule: Each winner is paid the highest cost that it could report and still remains a winner.

Now, we state our main results for TM-UNIFORM based on the non-uniform payment rule.

Theorem 2. TM-UNIFORM*, with modified non-uniform rate based threshold payment, is budget feasible, individually rational, truthful, and is* 3*-approximate compared to the optimum solution (which assumes access to the true costs).*

The proofs of Theorem 1 and Theorem 2 are presented in the longer version of the paper.

Randomized Mechanism (TM-RANDOMIZED)

In this section, we present a mechanism with an improved approximation ratio of $\frac{2e-1}{e-1} \approx 2.58$. We call this mechanism the *Randomized Uniform Mechanism* (TM-RANDOMIZED). Our mechanism is *truthful in large markets*, *i.e.*, as the market becomes larger:

- Extra utility that a person gains by misreporting her cost goes to zero.
- Ratio of any beneficial misreported cost to the true cost goes to one.

TM-RANDOMIZED produces a fractional matching (formally introduced below); if a fractional matching is not acceptable as the outcome of the mechanism, for example, if the tasks are not splittable, then we provide a way to convert (round) the produced fractional matching to an integral matching. The resulting mechanism produces an integral matching and it remains individually rational and truthful in large markets; also, the approximation ratio of $\frac{2e-1}{e-1} \approx 2.58$ still holds after this rounding. We now introduce some preliminaries and then describe the mechanism in detail.

Preliminaries

Recall $u_{\mathsf{max}} = \max_{t \in T} u_t$ and $\theta = \frac{u_{\mathsf{max}}}{U^*}$, that were used to model the large market assumptions.

Truthful Mechanisms in Large Markets: In our setting, we say that a mechanism is *truthful in large markets* when $\theta \to 0$ implies the following two properties for any $p \in P$:

- The extra utility gained by misreporting goes to zero. Formally, it means:

$$\left(\sup_{x} u_p(x, d_{-p}) - u_p(c_p, d_{-p}) \right) \to 0.$$

  where d_{-p} denotes any cost vector corresponding to the rest of players except p, and $u_p(x, d_{-p})$ denotes the utility of player p when he reports a cost x.
- The difference between a beneficial (misreported) cost and the true cost goes to zero. Formally, it means that:

$$|c_p - \overline{x}| \to 0 \quad \text{and} \quad |c_p - \underline{x}| \to 0.$$

  where

$$\overline{x} = \sup_{x} \{u(x, d_{-p}) \geq u(c_p, d_{-p})\},$$

$$\underline{x} = \inf_{x} \{u(x, d_{-p}) \geq u(c_p, d_{-p})\}.$$

Fractional Matchings: Let $m = |E(G)|$. A fractional matching $x \in \mathbb{R}^m$ is a vector that has an entry x_e for each edge e in G and satisfies the following conditions:

$$\sum_{t \in T} x_{(p,t)} \leq 1, \quad \forall p \in P \tag{1}$$

$$\sum_{p \in P} x_{(p,t)} \leq 1, \quad \forall t \in T \tag{2}$$

The utility of a fractional matching x is defined by

$$u(x) = \sum_{(p,t) \in E(G)} x_{(p,t)} \cdot u_t$$

The key concept in our randomized mechanism is a special fractional matching that we define as follows. For any graph $G(P,T)$ and permutation σ on the nodes of P, let $x(G,\sigma) \in \mathbb{R}^m$ denote the characteristic vector of the (integral) matching that is constructed by Procedure FindMatching(G,σ). Then, we define the fractional matching $x(G)$ as follows:

$$x(G) = \frac{1}{|P|!} \sum_{\sigma \in S_P} x(G,\sigma) \tag{3}$$

where S_P is the set of all permutations on the elements of P. Although we can not compute $x(G)$ in polynomial time, by sampling (a polynomial number of) many permutations, it can be computed with arbitrarily small error.

Description of the Mechanism

On the intuitive level, the mechanism does the following: it starts with a rate $r = \infty$ and computes the matching $x(G(r))$. If $r \cdot u(x(G(r))) > B$, then it slightly decrease the rate r. This is done until it reaches a rate $r = r^*$ such that $r^* \cdot u(x(G(r^*))) \leq B$. The mechanism stops at a rate r^* and produces a fractional matching x^*.

The allocation is defined by x^* in a natural way: Person p is assigned a fraction $x^*_{(p,t)}$ of each task $t \in T$. The payments are uniform: person p is paid $r \cdot u_t \cdot x^*_{(p,t)}$ for each task t.

Recall the sorted list of the edges, $e_1, \ldots, e_m$, in which the edges are sorted w.r.t. their buck per bang rate in decreasing order. Given this list, the mechanism is formally presented in Procedure TM-Randomized.

Procedure TM-Randomized

input : Graph $G(P,T)$, Budget B
output: A matching in G

$G' = G$;
for $i \leftarrow 1$ **to** m **do**
  $x = \frac{1}{|P|!} \sum_{\sigma \in S_P}$ FindMatching(G', σ);
  if $\mathsf{bb}(e_i) \cdot u(x) \leq B$ **then**
    $r \leftarrow \min\left(\frac{B}{u(\mathsf{x})}, \mathsf{bb}(e_{i-1})\right)$;
    break;
  $E(G') \leftarrow E(G') - \{e_i\}$;
Make the uniform payments;
Return x as the final matching;

We show that this mechanism is individually rational, truthful in large markets, and also, has approximation ratio $\frac{2e-1}{e-1}$. Theorem 3 states our main results for TM-RANDOMIZED, proof is presented in the longer version of the paper.

Theorem 3. TM-RANDOMIZED *is budget feasible, individually rational, truthful in large markets, and has approximation factor of* $\frac{2e-1}{e-1} \approx 2.58$ *compared to the optimum solution (which assumes access to the true costs).*

Indivisible Tasks and Rounding Procedure

If a fractional matching is not acceptable as the outcome of the mechanism, *e.g.*, tasks are not divisible, then we round the produced fractional matching to an integral matching. The resulting mechanism produces an integral matching and remains individually rational and truthful in large markets; furthermore, it has the same approximation ratio.

Formal details of the procedure are quite technical and involve understanding of the structure of the extreme points of the polytope corresponding to the budget feasible matchings – we leave it for the full version of paper. Below, we give a high level description of the rounding procedure.

We *round* the output of TM-RANDOMIZED, denoted by a fractional matching x, to an integral matching, *i.e.* we find integral matchings $x_1, \ldots, x_k$ and non-negative numbers $\lambda_1, \ldots, \lambda_k$ summing up to one such that $x = \sum_{i=1}^{k} \lambda_i x_i$. Moreover, we choose $x_1, \ldots, x_k$ such that they are almost budget feasible, *i.e.* $r \cdot u(x_i) \leq B + r \cdot u_{\mathsf{max}}$. Given such $x_1, \ldots, x_k$, we randomly choose one of them according to the probabilities given by $\lambda_1, \ldots, \lambda_k$.

In simple words, we can prove that a budget feasible fractional matching can be written as a convex combination of *integral* and *almost budget feasible* matchings. The rounding procedure outputs one of these matchings, each with probability equal to its coefficient in the convex combination.

This procedure outputs an almost budget feasible integral matching. To obtain a (strictly) budget feasible integral matching, we run the mechanism with a slightly reduced budget. This can be done without any (asymptotic) loss in the approximation ratio.

Extensions

We have presented our mechanisms for scenarios with one-to-one assignments, however, the mechanisms also work for finding many-to-many assignments. Here, we focus on the following two important extensions which can model many real-world applications, and in particular, are used to model the market in our experimental studies in next section.

- **Tasks can be done multiple times:** Consider the more general case when tasks can have decreasing reward functions. Let a task have reward r_i for being done in the i-th time, where $r_1 \geq \ldots \geq r_n$. We can reduce this to the basic setting by creating n identical copies of this task and defining a reward r_i for the i-th copy.
- **People can do multiple tasks:** If a person is willing to do up to d tasks, then we create d copies of this node and treat them as different individuals, *i.e.* each copy appears separately in the permutation σ.

All the properties that we proved for our mechanisms also hold in these extensions and the proofs are presented in the longer version of the paper. In this section, we briefly verify this fact for our simpler (non-randomized) mechanisms. The proofs for individual rationality and approximation ratio remain identical to the one-to-one setting. Also, for one-to-many assignments (where no person is assigned to more than a task), the proof for truthfulness remains the same. It remains to address truthfulness in the many-to-many setting.

Under the uniform payment rule, the mechanism remains one-way-truthful, the proof for this directly follows from the results of one-to-one assignment case. To get a fully truthful mechanism, we use the natural extension of the non-uniform payment rule for many-to-many assignments, and show that the mechanism remains truthful under this payment rule.

Payment Rule for Many-to-Many Assignments: Suppose person p is willing to do up to d tasks, which means he has d copies in the graph, namely $p_1, \ldots, p_d$. Then, p is paid $\sum_{i=1}^{d} \theta_i$, where θ_i is defined as follows: If copy p_i is assigned to no task by the mechanism, then $\theta_i = 0$, otherwise, θ_i is the highest cost that p could report such that p_i remains assigned to some task by the mechanism.

Next, we state our results for many-to-many assignments for TM-UNIFORM, proof is presented in the longer version of the paper.

Theorem 4. *Extension of* TM-UNIFORM *for many-to-many assignments is truthful under the non-uniform payment rule.*

Experimental Evaluation

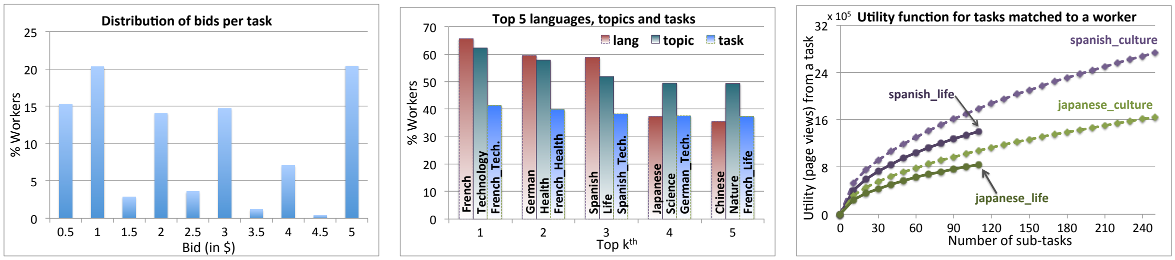
In this section, we carry out extensive experiments to understand the practical performance of our mechanism on simulated data, as well as on a realistic case study of translating popular Wikipeda pages to different languages using the MTurk platform. We begin by describing our experimental setup, benchmarks and metrics.

Experimental setup

Benchmarks: We compare our mechanism TM-UNIFORM against the following benchmarks and baselines:

- UNTM-GREEDY is an untruthful mechanism for matching which (unrealistically) assumes access to the true costs of the workers. It picks the edges iteratively in a greedy fashion based on maximal marginal value by cost ratio, and pay the worker the exact true cost. The mechanism runs until the budget is exhausted. This is a two factor approximation of the OPT, the untruthful optimal solution (Goel and Mehta 2008).
- UNTM-RANDOM is a trivial untruthful mechanism for matching that picks the edges (an available worker-task pair) in a random order iteratively, paying the exact cost to the worker.
- TM-MEANPRICE is a trivial truthful mechanism for matching which picks the edges randomly (same as in UNTM-RANDOM), however it offers a fixed price payment (set to be the *mean* of the whole set of workers in the crowdsourcing market). If the payment is higher than current worker's cost, the worker would accept the offer, otherwise rejects. This serves as a trivial lower bound baseline for our mechanism TM-UNIFORM. This baseline reflects the kind of pricing strategies often used by job requesters on online platforms like MTurk.

Metrics and experiments: The primary metric we track is the utility of the mechanism for a given budget. On synthetic data, we vary the amount of available budget to see the effect on our mechanism. We also vary the degree of graph connectivity between tasks and workers to understand the effect of matching constraints. Additionally, we vary the variance of



(a) Distributions of Bids ($) (b) Top languages, topics and tasks (c) Worker's profile

Figure 1: (a) Distribution of workers' bids ($), **(b)** Top languages, topics and tasks for MTurk workers, and **(c)** Illustrates a profile of worker who picked *Spanish* and *Japanese* as target languages of interest; along with *Culture* and *Life* as topics of interest. This corresponds to total of four types of tasks that this worker can perform, namely: *Spanish_Culture*, *Spanish_Life*, *Japanese_Culture* and *Japanese_Life*. Topic *Culture* contains 253 pages and topic *Life* contains 120 pages, which decide the total number of sub-taks available in these tasks. The concave utility function for each task is obtained by sorting the pages in decreasing order of utility and summing it up. Further, *Spanish* internet users' population is 164.9 million, compared to 99.2 million for *Japanese* which dictates the scale of 1.66 between the graphs of *Spanish_Culture v.s Japanese_Culture* and also between *Spanish_Life v.s Japanese_Life*.

tasks' utilities and workers' cost to understand its impact, specially on the truthful mechanisms. In our experiments on MTurk data, the utility is directly mapped to the absolute number of page views from the tasks completed by the mechanism and budget directly maps to the amount of available money in U.S. dollars ($) that can be spent for crowdsourcing. Apart from the overall utility, we further track the utility acquired per target language and source topic. Our main goal is to gain insights into the execution of the mechanisms which arise from the market dynamics (*e.g.*, high availability or shortage of workers with specific skills as well as the utility difference between different type of tasks).

Distributions and parameter choices: For synthetic experiments, we considered a simple market, where each worker can do only one task and each task can be done only once. We used uniform distribution with range of $[0.1 - 0.9]$ to generate the tasks' utilities as well as the workers' costs. We generated a random graph with 200 workers, 200 tasks and a probability of edge formation being set to 0.3. We further vary these ranges of the distribution and graph degree in the experiments below. For the real-world experiments on Wikipedia translation case study, the data was collected from online resources and MTurk as is further described in detail below. We didn't perform any specific scaling or normalization of the values for real study, so as to make the utility acquired easily interpretable from an actual application point of view (*e.g.*, page view counts for a given budget in US dollars $). Next, we describe in detail the process of gathering real data for our experiments.

Wikipedia translation on Mechanical Turk

We now describe our Wikipedia translation project in detail including data collection from online resources and workers' preference elicitation from MTurk platform.

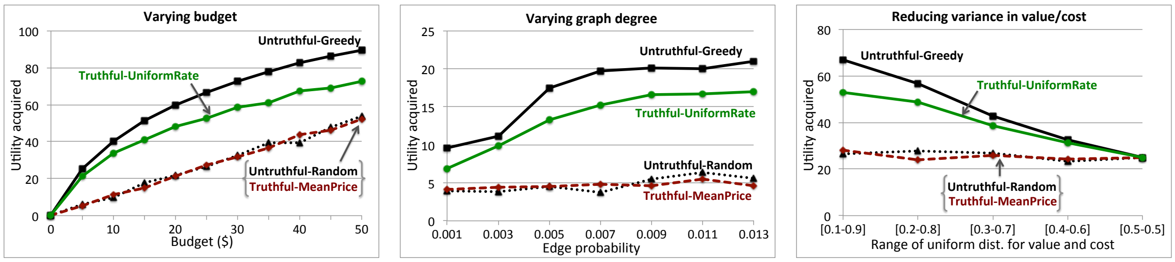
Case study on Wikipedia translation project. Our experiments are inspired by the application of translating Wikipedia's popular or trending articles to other languages, making them easily accessible to every internet user. We intend to use crowdsourcing for this application, where different workers can manage or perform the translation tasks, possibly with help of available software tools. More concretely, our goal is to translate the weekly top 5,000 most popular pages of English Wikipedia to the top ten most widely used languages on internet. Here, a task heterogeneity comes from the topic of the page and the target language. As workers could have different topical interests and different expertise or preference for the target languages, this creates the need for matching the right set of workers for the tasks they can perform. We considered total of 25 different topics based on the top level classification topics actually used in Wikipedia[5]. Next we considered the 10 most widely used internet languages (after English), along with their user base on internet[6,7]. These together gives us a total of 250 different heterogeneous tasks (25 topics times 10 languages). Next, we obtained the list of top 5,000 pages from Wikipedia for one of the weeks in September 2013[8] along with their page view count. We then annotated each one of these pages to one of the 25 topics. Instead of using some classifier or inferring top level topic from Wikipedia's taxonomy, we resorted directly to MTurk to obtain this annotation. We posted a Human Intelligence Task (HIT) which asked workers to annotate each one of these pages with unique topic from the list of 25 provided to them. At the end of this whole process, we have a set of 250 heterogeneous tasks associated with a topic and target language. Each task can further be done multiple times, which equals the number of pages annotated with the topic of this task – we refer to them as *sub-tasks*. The utility associated with a sub-task is simply obtained by multiplying user base of target language and page view count of the page (this simply denotes the effective page view count the application will have from this subtask). These utilities for all the sub-tasks (ordered in their decreasing value) of a task form the concave utility curve associated with the task. This is illustrated in Figure 1(c).

[5]http://en.wikipedia.org/wiki/Category:Main_topic_classifications

[6]http://en.wikipedia.org/wiki/Languages_used_on_the_Internet

[7]http://pocketcultures.com/topicsoftheworld/files/2011/09/Internet-Language-Infographic.png

[8]http://en.wikipedia.org/wiki/User:West.andrew.g/Popular_pages



(a) Varying budget ($) (b) Varying graph degree (c) Varying value/cost variance

Figure 2: Results for experiments on synthetic data. **(a)** Overall utility acquired by varying budget.TM-UNIFORM performance is within a margin of 20% compared to that of UNTM-GREEDY (which assumes unrealistic access to true costs). TM-UNIFORM shows up to 100% improvement over TM-MEANPRICE, a typical fixed price mechanism used by requester on crowdsourcing platforms like MTurk. **(b)** Utility acquired as degree of graph is varied, for a fixed budget of 5$. **(c)** Effect of varying value/cost variance in the market, by reducing the range of uniform distribution used for sampling task's utilities and worker's costs. The results illustrate that markets with higher variance increases the strategic power of the workers.

MTurk data and worker's preferences. Next, our goal was to infer worker preferences in terms of topical interests as well the target languages they are interested in. We posted a HIT on MTurk platform in form of a survey, where workers were told about an option to participate in our research prototype of Wikipedia translation project. Our HIT on MTurk stated the survey's purpose as to understand the feasibility of our project, requesting workers to provide correct and honest information. We clearly stated that workers are not required to know the target language at this point and they can potentially be trained with set of tools to assist in our translation project. Our survey explicitly asked following questions to the workers:

- Choose up to 10 topics from the list below based on your interests for the source pages of the tasks you would be interested to perform.
- Choose up to 5 languages from the list below based on your interests for the target languages of the tasks you would be interested in owning.
- Roughly, from 0.1$ to 5$, what price would you like to receive per task?
- Roughly, from 1 to 100, how many tasks would you like to perform per week?

Given the preference information elicited from this HIT, we defined a skill for worker as combination of preference of page topic and target language. This, together with the characterization of the tasks, provides us with the graph of matching constraints between workers and tasks.

Statistics A total of 1000 workers participated in our survey. We didn't restrict our survey to any geographical region, to allow for maximal variability in our study given the nature of the application. Figure 1(a) shows the distribution of bids collected. Figure 1(b) shows the top five languages and topics which were preferred by workers. Figure 1(b) also illustrates the percentage of workers who can perform a particular type of task based on the inferred matching constraints. Figure 1(c) shows the profile of a worker who picked *Spanish* and *Japanese* as languages; along with *Life* and *Culture* as topics. This worker can do a total of four different type of tasks, as illustrated in Figure 1(c) along with their utility curves inferred from the associated sub-tasks.

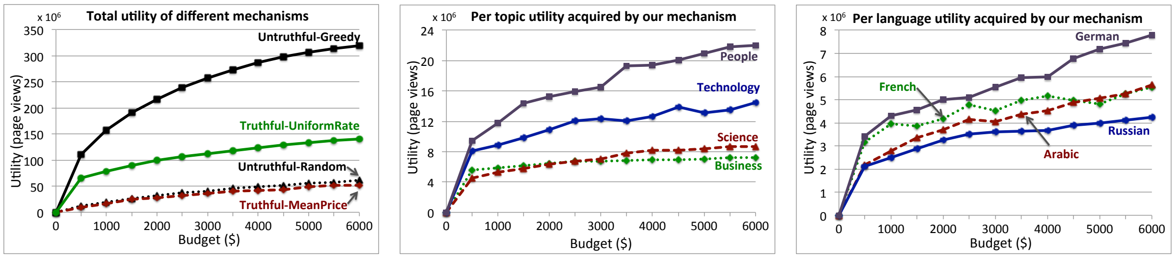
Results on Synthetic data

We now discuss the findings from our experiments, starting with results on synthetic data.

Varying budget. Figure 2(a) shows the utility acquired by different mechanisms as we vary the available budget. On synthetic data, our truthful mechanism TM-UNIFORM performs within a margin of 20% compared to that of UNTM-GREEDY with (unrealistic) access to true costs. Both the trivial baselines for untruthful mechanism UNTM-RANDOM and truthful mechanism TM-MEANPRICE perform relatively worse. We note that the very similar performance of UNTM-RANDOM and TM-MEANPRICE is actually attributed to the fact that our cost and value distributions on which results are reported here are uniform. A skewed distribution or using a different fixed price for TM-MEANPRICE (for example, median of worker's population) could perform better or worse compared to UNTM-RANDOM. However, both these mechanisms come without any guaranteess and can perform arbitrarily bad, as we will see on real data experiments.

Varying graph degree between workers and tasks. We vary the degree of connectedness between workers and tasks, which in turn could affect the availability of skills in workers pool for a given task, affecting the performance of the mechanisms. Figure 2(b) studies this for a fixed budget of 5$. Starting from a very low connectivity of 0.001, we increment it in steps to see the affect on acquired utility. Both TM-UNIFORM and UNTM-GREEDY show an increasing performance with saturated gains, though the naive mechanisms UNTM-RANDOM and TM-MEANPRICE almost remain stagnant in terms of their performance.

Varying value/cost variance in market. Another aspect we study on the synthetic data is the variance in utility of tasks and that of workers costs, as illustrated in Figure 2(c). As expected, in the extreme case of no variance, all the mechanisms perform the same. And, as variance in market increases, the relative performance of our mechanism TM-MEANPRICE decreases w.r.t UNTM-GREEDY. Intuitively, this shows that the markets with higher variance results in increasing the strategic power of the workers.



(a) Overall utility (b) Utility per topic (c) Utility per language

Figure 3: Results for experiments on Wikipedia translation using MTurk. **(a)** Overall utility acquired by varying budget.TM-UNIFORM performance is within a margin of 55% compared to that of UNTM-GREEDY (which assumes unrealistic access to true costs). And, we see up to 100% improvement over TM-MEANPRICE. **(b)** and **(c)** illustrates market dynamics by showing the utility acquired per different topic and language as budget is varied. In **(c)**, *French* language acquires higher utility in the beginning, attributed to bigger pool of available workers (65.7% for *French* vs 27% *Arabic* on MTurk). Eventually *Arabic* language catches up because of higher utilities associated with sub-tasks attributed to larger user base of the language (59.8 million for *French* vs 65.4 million for *Arabic*).

Results on Wikipedia translation data

Next, we measure the performance of our mechanisms on the real world data gathered as part of Wikipedia translation project using MTurk workers.

Varying budget. Figure 3(a) illustrates the results of utility on real data. Here, the utility corresponds directly to the page-view counts that mechanism would generate on internet and budget corresponds to US dollars ($) we are given. The utility of TM-UNIFORM is about 55% lower than UNTM-GREEDY, worse than what we observed on synthetic data (20% lower). This is because of higher variance of task values and worker's costs in real data, increasing the strategic power of workers and affecting the performance of truthful mechanisms (see also Figure 2(c)). And, we see up to 100% improvement over TM-MEANPRICE. The fixed price mechanisms like TM-MEANPRICE are often used by requesters currently in online crowdsourcing platforms like MTurk. The performance of our mechanism TM-UNIFORM compared to TM-MEANPRICE shows the potential gains we can expect by using our mechanisms in current crowdsourcing platforms.

Utility acquired per topic. Next, we study the effect of market dynamics in a real crowdsourcing market. Figure 3(b) shows the utility acquired per topic as we vary the budget. For a given topic, the acquired utility depends on the number of workers interested in the topic as well as the page view count of pages which fall in these topics. For example, some pages related to recent sports events (in topic *Sports*) or entertainment pages (in topic *Arts*) could be much more popular compared to a page, let's say, in topic *Law*. Figure 3(b) shows the utility of four topics *People*, *Technology*, *Science* and *Business* as the budget is varied. The dynamics can be seen between *Science* and *Business* – topic *Business* acquires higher utility in the beginning because of presence of some highly visited pages which fall in this category. However, *Science* quickly takes over as the pool of MTurk workers interested in *Science* topic is much larger than that for *Business* (49.50% compared to 34.72%).

Utility acquired per language. Along the same lines as above, Figure 3(c) illustrates the results for utility per lanaguge. We plot the results for four languages: *German*, *French*, *Arabic* and *Russian*. The dynamics of acquired utility for a language are controlled by corresponding user base on internet which is 75.5 Million(M), 59.8 M, 65.4 M and 59.7M respectively for these languages. Additionally, the interests of MTurk workers affect the availability of worker pool which in our data corresponds to 59.6%, 65.7%, 27% and 34.7%, respectively. The *French* language acquires higher utility in the beginning, attributed to bigger pool of available workers on MTurk. Eventually, *Arabic* language catches up because of higher utilities associated with subtasks attributed to larger user base of the language.

Conclusions and Future Work

In this paper, we studied the mechanism design problem for crowdsourcing markets with matching like constraints, inspired by the realistic crowdsourcing project of translating Wikipedia articles. We designed mechanisms with strong theoretical guarantees which are complemented by extensive experimentation to show their real-world applicability.

There are some natural extensions for future work. We assumed that workers have same cost for all the tasks. Extending our mechanisms to general setting where workers can have different costs for different tasks would be practically useful. Another interesting generalization would be when tasks require multiple workers for them to be finished.

We would like to point out that our mechanism TM-UNIFORM takes as input a permutation on the workers. This extra input gives a useful tool to manipulate the outcome of the mechanism. For instance, if some workers are more desired over others (say, based on quality ratings or demographics information), one can put these workers in the front of the permutation. Lastly, our approximation factor 3 works for the worst case permutation. It is an open question if one can show better guarantees on a randomly selected permutation. Our conjecture is that the expected performance of the uniform mechanism on a randomly selected permutation will be same as our randomized mechanism.

Acknowledgments. Adish Singla is supported by Nano-Tera.ch program as part of the Opensense II project.

References

[2011] Aggarwal, G.; Goel, G.; Karande, C.; and Mehta, A. 2011. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proc. of SODA*, 1253–1264. SIAM.

[2012] Badanidiyuru, A.; Kleinberg, R.; and Singer, Y. 2012. Learning on a budget: Posted price mechanisms for online procurement. In *EC*.

[2013] Badanidiyuru, A.; Kleinberg, R.; and Slivkins, A. 2013. Bandits with knapsacks. In *FOCS*.

[2012] Bei, X.; Chen, N.; Gravin, N.; and Lu, P. 2012. Budget feasible mechanism design: from prior-free to bayesian. In *STOC*.

[2011] Chen, N.; Gravin, N.; and Lu, P. 2011. On the approximability of budget feasible mechanisms. In *SODA*. SIAM.

[2009] Devanur, N. R., and Hayes, T. P. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *ACM Conference on Electronic Commerce*.

[2013] Devanur, N. R.; Jain, K.; and Kleinberg, R. D. 2013. Randomized primal-dual analysis of ranking for online bipartite matching. In *SODA*. SIAM.

[2013] Difallah, D. E.; Demartini, G.; and Cudré-Mauroux, P. 2013. Pick-a-crowd: tell me what you like, and i'll tell you what to do. In *WWW*, WWW '13, 367–374.

[2008] Goel, G., and Mehta, A. 2008. Online budgeted matching in random input models with applications to adwords. In *SODA*. SIAM.

[2012] Ho, C.-J., and Vaughan, J. W. 2012. Online task assignment in crowdsourcing markets. In *AAAI*.

[2010] Horton, J. J., and Chilton, L. B. 2010. The labor economics of paid crowdsourcing. In *EC*, 209–218.

[2010] Horton, J. J., and Zeckhauser, R. J. 2010. Algorithmic wage negotiations: Applications to paid crowsourcing. In *CrowdConf*.

[1990] Karp, R. M.; Vazirani, U. V.; and Vazirani, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *STOC*.

[2007] Mehta, A.; Saberi, A.; Vazirani, U.; and Vazirani, V. 2007. Adwords and generalized on-line matching. In *Journal of the ACM, Volume 54 Issue 5*.

[2013] Singer, Y., and Mittal, M. 2013. Pricing mechanisms for crowdsourcing markets. In *WWW*, 1157–1166.

[2010] Singer, Y. 2010. Budget feasible mechanisms. In *FOCS*, 765–774.

[2011] Singer, Y. 2011. How to win friends and influence people, truthfully: Influence maximization mechanisms for social networks. In *WSDM*. ACM.

[2013a] Singla, A., and Krause, A. 2013a. Incentives for privacy tradeoff in community sensing. In *AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*.

[2013b] Singla, A., and Krause, A. 2013b. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *WWW*, WWW '13, 1167–1178.

[2010] Tran-Thanh; L.; A., C.; de Cote, J. E. M.; Rogers, A.; and Jennings, N. R. 2010. Epsilon-first policies for budget-limited multi-armed bandits. In *AAAI*, 1211–1216.

[2012a] Tran-Thanh, L.; Chapman, A.; Rogers, A.; and Jennings, N. R. 2012a. Knapsack based optimal policies for budget-limited multi-armed bandits. In *AAAI*.

[2012b] Tran-Thanh, L.; Stein, S.; Rogers, A.; and Jennings, N. R. 2012b. Efficient crowdsourcing of unknown experts using multi-armed bandits. In *AAAI*.

Appendix: Proof of Theorem 1

In order to prove Theorem 1, we first prove the following properties of the mechanism, namely, individual rationality in Lemma 1 and oneway-truthfulness in Lemma 3.

Lemma 1. *The uniform mechanism is individually rational.*

Proof of Lemma 1. Individual rationality follows from the fact that the buck per bang rate of each edge in G^* is at most r^*. More precisely, for each edge $(p,t) \in E(G^*)$ we have that $c_p/u_t \leq r^*$. Now, suppose p is assigned a task by the mechanism, then since $(c_p, u_{\mathsf{M}(p)}) \in E(G^*)$, we have $c_p \leq r^* \cdot u_{\mathsf{M}(p)}$. Observing that $r^* \cdot u_{\mathsf{M}(p)}$ is exactly the payment of the mechanism to p finishes the proof. □

Next, we prove a lemma which will be used to prove oneway-truthfulness in Lemma 3.

Lemma 2. *If a person p reports a cost higher than c_p, then* TM-UNIFORM *assigns her either the same task (as if she has reported c_p) or no task at all.*

Proof of Lemma 2. Consider a scenario where a person $p \in P$ lies by reporting a higher cost $\overline{c_p}$ instead of c_p; we denote this new instance (in which c_p is replaced by $\overline{c_p}$) by the *fake instance*. Also, let (r^*, e) and $(\overline{r^*}, \overline{e})$ respectively denote the iteration at which the mechanism stops in the real and fake instance. The proof has two cases: either $r^* \leq \overline{r^*}$ or $r^* > \overline{r^*}$. Later, we will show that the second case never happens. Now we prove the lemma in the first case.

Case 1 If $r^* \leq \overline{r^*}$, then let $H, \overline{H}$ respectively denote G' at iteration $(\overline{r^*}, \overline{e})$ in the real and fake instance. Also, denote the matchings produced by running Procedure FindMatching on $H, \overline{H}$ respectively by $\mathsf{M}, \overline{\mathsf{M}}$. Note that $\overline{\mathsf{M}}$ is in fact the matching produced by TM-UNIFORM in the fake instance.

We will prove that either $\overline{\mathsf{M}} = \mathsf{M}$ or $\overline{\mathsf{M}}(p) = \emptyset$; this would contradict with p having incentive to report a fake cost and would prove the lemma for the current case, i.e. $r^* \leq \overline{r^*}$. To this end, let $t_1, \ldots, t_k$ be the set of tasks to which p is interested (connected) in H, such that $u_{t_1} \geq \ldots \geq u_{t_k}$. For expositional simplicity in presenting the proof, we also add a dummy task $t_{k+1} = \emptyset$ where $u_{t_{k+1}} = 0$. Assigning t_{k+1} to p means assigning no task to p.

Since $\mathsf{M}(p) \in \{t_1, \ldots, t_k\}$, there exists some i such that $\mathsf{M}(p) = t_i$. Since p has reported a higher cost, a (possibly empty) subset of its neighbors in H become unavailable in $\overline{H}$. This subset must be of the form $\{t_j, \ldots, t_k\}$ (where by convention, we suppose having $j > k$ means that the subset is empty). If $j \leq i$, then we must have $\overline{\mathsf{M}}(p) = \emptyset$, because otherwise, the mechanism would have assigned $\overline{\mathsf{M}}(p)$ to p in the real instance. If $j > i$, then t_i is still available for p in $\overline{H}$, which means $\overline{\mathsf{M}}(p) = t_i = \mathsf{M}(p)$. This implies $\overline{\mathsf{M}} = \mathsf{M}$. To summarize, we showed that either $\overline{\mathsf{M}} = \mathsf{M}$ or $\overline{\mathsf{M}}(p) = \emptyset$ hold, which is a contradiction and proves the lemma in the first case.

Case 2 It remains to show that the second case, $\overline{r^*} < r^*$, never happens. Let $H, \overline{H}$ respectively denote G' at iteration (r^*, e) in the real and fake instance. Also, let $\mathsf{M}, \overline{\mathsf{M}}$ respectively be the matchings produced by running Procedure FindMatching on $H, \overline{H}$. We will prove that $u(\overline{\mathsf{M}}) \leq u(\mathsf{M})$ which is a contradiction: it implies $r^* \cdot u(\overline{\mathsf{M}}) \leq B$, which means the stopping rate in the fake instance must have not been smaller than r^*.

Claim 1. $u(\overline{\mathsf{M}}) \leq u(\mathsf{M})$.

Proof. To prove this claim, again let $t_1, \ldots, t_k$ be the set of tasks to which p is interested in $\overline{H}$, such that $u_{t_1} \geq \ldots \geq u_{t_k}$. If p is assigned to a task t_i in $\overline{\mathsf{M}}$ with $1 \leq i \leq k$, then it must be assigned to the same task in M: it will be the highest available task that FindMatching can assign to it. Since p is assigned to the same task in M and $\overline{\mathsf{M}}$, then we have $\mathsf{M} = \overline{\mathsf{M}}$. So, assume that p is not assigned to any of the tasks $t_1, \ldots, t_k$, which also means that it is not assigned to any tasks in $\overline{\mathsf{M}}$. Now, consider the graph $\mathsf{M} \triangle \overline{\mathsf{M}}$. This graph has a single non-empty component, which is a path Q that has p as one of its endpoints. It is straight-forward to verify that this means $u(Q \cap \overline{\mathsf{M}}) \leq u(Q \cap \mathsf{M})$, which implies $u(\overline{\mathsf{M}}) \leq u(\mathsf{M})$. □

However, as we mentioned earlier, having $u(\overline{\mathsf{M}}) \leq u(\mathsf{M})$ gives a contradiction and proves the lemma in the second case as well. □

Lemma 3. *The uniform mechanism is oneway-truthful.*

Proof of Lemma 3. This follows from Lemma 2. □

We now prove Theorem 1 by using the above results.

Proof of Theorem 1. Individual rationality and oneway-truthfulness are implies by Lemma 1 and Lemma 3. It remains to prove the approximation ratio. Let U be the total utility achieved by the uniform mechanism and U^* denote the optimal utility that we could achieve (if the costs where known) with budget B. First, note that although the uniform mechanism is only oneway-truthful (and not fully truthful), the analysis can go forward as if we are analyzing a truthful mechanism: in a oneway-truthful mechanism, players have incentive to report only lower costs; this can change the value of the optimum solution, but note that it can only increase it to a larger value $\overline{U^*}$. If we prove that our mechanism is a 3-approximation for the reported instance, then we have shown that it achieves utility at least $\overline{U^*}/3$, which is also at least $U^*/3$, i.e. it would be a 3-approximation for the real instance too.

The proof has two cases: we either spend the whole budget, or not. First we give a proof for the case when we spend the whole budget. Also, let the iteration at which the mechanism stops be denoted by (r, e).

Case 1 We decompose $E(G)$ to two subsets, E_r^- and E_r^+. Subset E_r^- contains edges which were still in G' when the mechanism stopped (to clarify, note that all the edges in E_r^- have buck per bang rate at most r, but there might be edges with rate exactly r which are not in E_r^-). Let E_r^+ contain the rest of the edges in $E(G)$.

Then, define $U^* = U^- + U^+$ where U^-, U^+ respectively denote the portion of the utility in the optimal solution which is gained from E_r^-, E_r^+. We prove that $U^+ \leq U$ and $U^- \leq 2U$. Consequently, we would have $U^* \leq 3U$ which proves the lemma.

To see $U^+ \leq U$, note that $U^+ \leq B/r$ since all the edges in E_r^+ have buck per bang at least r. Also, note that $B/r = U$, since in the mechanism, we pay r for each unit of utility and spend the whole budget. This implies $U^+ \leq U$.

To see $U^- \leq 2U$, just note that Procedure FindMatching is the greedy algorithm for finding a maximum matching, and it is a well-known fact that the greedy algorithm has approximation factor 2. This proves the lemma in Case 1.

Case 2 Now we prove the lemma in the case when we do not spend the whole budget. We follow the proof for the previous case with a slight difference: we show that the left over budget is very small (instead of being 0). In fact, if we show that the left over budget is at most $r u_{\mathsf{max}}$, then instead of having $U^+ \leq U$, we would have $U^+ \leq U + u_{\mathsf{max}}$, and everything else in the proof remains the same. Consequently, we would have $U^* \leq (3 + o(1)) \cdot U$.

To this end, we closely follow what happens in the mechanism in the case when we have left over budget. Let $H, \overline{H}$ respectively denote subgraphs of G with the edge sets E_r^- and $E_r^- \cup \{e\}$, where, recall that e is the last edge that was removed from G' in the mechanism. The mechanism found a matching $\overline{M}$ in $\overline{H}$, which was not budget feasible with rate $r = \mathsf{bb}(e)$. After removing e, the mechanism found a matching M which is budget feasible with the same rate r. This is the only scenario which leads to a left over budget in the mechanism.

The key to bound the left over budget is that $u(\mathsf{M})$ and $u(\overline{\mathsf{M}})$ are roughly the same. Particularly, it is straight-forward to verify that adding a single edge to H (here, the edge e) can not increase the utility of the output of FindMatching by more than u_{max}. So, we have $u(\overline{\mathsf{M}}) \leq u(\mathsf{M}) + u_{\mathsf{max}}$. This fact, and the fact that $r \cdot u(\mathsf{M}) < B < r \cdot u(\overline{\mathsf{M}})$ together imply that the left over budget is at most ru_{max}. □

Appendix: Proof of Theorem 2

To prove the results in Theorem 2 for TM-UNIFORM based on the non-uniform payment rule, we only need to prove the truthfulness and budget feasibility. The efficiency and approximation factor remains same, as the allocation is same. Below, we prove truthfulness of the mechanism in Lemma 4.

Lemma 4. *Mechanism TM-Uniform is truthful under the Non-uniform Payment Rule.*

Proof of Lemma 4. Proof by contradiction. Suppose person p has incentive to lie. First, verify that p can not be a winner since winners have no incentive to lie due to the payment rule. So, assume p is not a winner and has incentive to create a fake instance by reporting a cost $\overline{c_p} \neq c_p$. Now, recall that p could not win when she reports c_p, or a cost higher than c_p. So, the payment to p is bounded by c_p, which means, even if she misreports and wins, her utility will be at most zero. So, p has no incentive to lie. □

Next, we prove its budget feasibility, *i.e.*, the the payments will not exceed the budget in Lemma 5 below.

Lemma 5. *Mechanism TM-Uniform is budget feasible with the Non-uniform Payment Rule.*

Proof of Lemma 5. Let M denote the matching produced in the last iteration of the mechanism. We show that the payment to person p is not higher than the payment she receives under the uniform payment rule. This will imply budget feasibility of the non-uniform payment rule due to budget feasibility of the uniform payment rule.

Let (r^*, e) denote the iteration at which the mechanism stops. As we mentioned above, we prove that the payment to p is not more than $r^* \cdot u(\mathsf{M}(p))$ (the payment she receives under the uniform payment rule). Now, for contradiction, assume that p can report a cost $\overline{c_p}$ which is larger than $r^* \cdot u(\mathsf{M}(p))$ and still remains a winner. Call the instances in which p reports costs c_p and $\overline{c_p}$ as the small and large instances, respectively. Also, let $(\overline{r^*}, \overline{e})$ denote the iteration at which the mechanism stops in the large instance. We will show that if $(r^*, e) \neq (\overline{r^*}, \overline{e})$ then we get a contradiction. On the other hand, having $(r^*, e) = (\overline{r^*}, \overline{e})$ implies $\overline{c_p} \leq r^* \cdot u(\mathsf{M}(p))$, (since p is a winner in the large instance) which would be a contradiction again. To prove $(r^*, e) \neq (\overline{r^*}, \overline{e})$, first we show that $r^* \leq \overline{r^*}$.

Claim 2. $r^* \leq \overline{r^*}$.

Proof. Proof by contradiction. Suppose $r^* > \overline{r^*}$. Then, consider when the mechanism reaches to buck per bang rate r^* in the small instance. At this rate, Procedure FindMatching did find a matching M in the small instance for which $r^* \cdot u(\mathsf{M}) \leq B$. Now, see that at rate r^*, FindMatching also finds a matching $\overline{\mathsf{M}}$ in the large instance, for which $u(\overline{\mathsf{M}}) \leq u(\mathsf{M})$. It is straight-forward to verify this fact since we have $\overline{c_p} > c_p$.

Finally, observe that $u(\overline{\mathsf{M}}) \leq u(\mathsf{M})$ and $r^* \cdot u(\mathsf{M}) \leq B$ imply $r^* \cdot u(\overline{\mathsf{M}}) \leq B$. This means the mechanism had to stop at rate r^* in the large instance since it finds a feasible matching. Contradiction. □

So, assume $r^* \leq \overline{r^*}$. Now, let $\mathsf{N}, \overline{\mathsf{N}}$ denote the matchings that the mechanism produces at iteration $(\overline{r^*}, \overline{e})$ respectively in the small and large instance. We show that $\mathsf{N} = \overline{\mathsf{N}}$, which is a contradiction since it means the mechanism should have stopped at iteration $(\overline{r^*}, \overline{e})$ in the small instance.

While running the mechanism on the small instance, consider when it reaches to iteration $(\overline{r^*}, \overline{e})$. At this point, the task $\overline{\mathsf{N}}(p)$ must be available for p also in the small instance. Moreover, this is the task with the highest utility which is available for p (otherwise the mechanism would have chosen a different match for p in the large instance). Consequently, at iteration $(\overline{r^*}, \overline{e})$, person p will be matched to $\overline{\mathsf{N}}(p)$ in the small instance as well, and so, the outcome of the mechanism in the small instance would be identical to the outcome of the mechanism in the large instance. Contradiction. □

We now prove the main result Theorem 2 by using the above results.

Proof of Theorem 2. The truthfulness and budget feasibility follows from the results of Lemma 4 and Lemma 5. The efficiency and approximation factor follows from results of Theorem 2, as the allocation is same. □

Appendix: Proof of Theorem 3

Proof of Theorem 3. We defer the proofs for individual rationality and truthfulness to the full version of the paper. Here, we only prove the approximation ratio. The idea is that although Procedure FindMatching is only a 2-approximation for a fixed permutation σ, it becomes a $\frac{e}{e-1}$-approximation if permutation σ is chosen uniformly at random. This holds due to Section 3 of (Goel and Mehta 2008).

Given this fact, we follow the proof of Theorem 1. Let U be the total utility gained by the uniform mechanism and U^* denote the optimal utility that we could gain with budget B. Also, let the stopping time of the mechanism be denoted by (r, e). Decompose $E(G)$ to two subsets, E_r^- and E_r^+. Subset E_r^- contains edges with buck per bang at most r which were in G' when the mechanism stopped. Let E_r^+ contain the rest of the edges in $E(G)$.

Then, define $U^* = U^- + U^+$ where U^-, U^+ respectively denote the portion of the utility in the optimal solution that is achieved from E_r^-, E_r^+. We prove that $U^+ \leq U + u_{\mathsf{max}}$ and $U^- \leq \frac{e}{e-1} \cdot U$, which means $U^* \leq \left(\frac{2e-1}{e-1} + o(1)\right) \cdot U$.

The proof for $U^+ \leq U + u_{\mathsf{max}}$ is identical to Theorem 1. It remains to show $U^- \leq \frac{e}{e-1} \cdot U$. This follows from Section 3 of (Goel and Mehta 2008): Procedure FindMatching is a $\frac{e}{e-1}$-approximation if permutation σ is chosen uniformly at random. So, if we run it on the graph $G^-(P, T)$ with $E(G^-) = E_r^-$, then it outputs a matching with expected utility at least $\frac{e-1}{e} \cdot U^-$. On the other hand, the expected utility of the matching is, by definition, equal to U, i.e. the utility of the fractional matching that we construct in TM-Randomized by taking the average over all permutations σ. This just means $\frac{e-1}{e} \cdot U^- \leq U$. □

Appendix: Proof of Theorem 4

Proof of Theorem 4. Fix a person p and suppose she is willing to do up to d tasks, which means she has d copies in the graph, namely $p_1, \ldots, p_d$. We prove that for any copy p_i, the maximum possible utility that person p can derive from p_i (even by misreporting) is $\max\{0, \theta_i - c_p\}$ (where utility of a person is defined to be the payment she receives minus her true cost). This fact implies that person p has no incentive to misreport: because for any copy p_i, the mechanism is providing her the maximum possible utility that she can ever achieve from that copy, i.e. $\max\{0, \theta_i - c_p\}$.

It remains to show that the maximum possible utility that person p can derive from p_i (even by misreporting) is $\max\{0, \theta_i - c_p\}$. First, we verify this fact for when p_i is assigned to some task: then, by definition, θ_i is the highest payment that p can ever receive for copy p_i; i.e. she will not get paid more that θ_i for p_i by misreporting. This just means her utility is bounded by $\max\{0, \theta_i - c_p\}$.

On the other hand, if p_i is assigned to no tasks, then by the definition of the payment rule, p will never get paid more than c_p for copy p_i (even by misreporting). So, her utility from p_i will never be more than $0 = \max\{0, \theta_i - c_p\}$. □