# Matching Workers Expertise with Tasks: Incentives in Heterogeneous Crowdsourcing Markets *

**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 the online crowdsourcing markets. In this paper, we consider the following realistic setting of online crowdsourcing markets - we are given a heterogeneous set of 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 on a realistic case study of Wikipedia translation project using Mechanical Turk. Our results demonstrate the practical applicability of our mechanisms for realistic crowdsourcing markets on the web.

We note that this is the first paper that addresses this setting of matching tasks to workers from a mechanism design perspective. Previous work either made a simplifying assumption that tasks are homogeneous or didn't consider the matching constraints given by the bipartite graph.

## 1 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 increased 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], ClickWorker [2] and CrowdFlower [3]). The crowdsourcing tasks are of variety of nature including image annotation, rating search engine results, 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. Because of the budget constraints, pricing the tasks too high can result in lower output for the requester. On the other hand, pricing the tasks too low can disincentivize workers. This trade-off between efficiency and workers' incentives make the pricing decisions in crowdsourcing markets complex, and 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.

*Under review by the International World Wide Web Conference (WWW) 2014

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 [4, 5, 6, 7, 8] have proposed the use of budget feasible procurement auction (first introduced by [4]) 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.

### 1.1 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 the 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 task, 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 simplification, let us assume that a worker can do only one task (we will relax this constraint later). The goal is to design an auction mechanism that - i) is incentive compatible in the sense that it is truthful for agents to report their true cost, ii) picks a set of workers and assigns them to the tasks they are eligible for in order to maximize the utility of the requester, iii) makes sure total payments made to the workers doesn't exceed the budget of the requester. We deviate from previous works as we look at a more general and realistic case of assigning workers to tasks that match their skill sets.

In Section 4, we present a novel mechanism TM-UNIFORM (*i.e.* Truthful Matching using Uniform Rate), which allocates the tasks to workers under the matching constraints; is incentive-compatible, budget feasible and obtains utility with approximation ratio 3 *w.r.t* optimal (with full knowledge of true costs). In Section 5, we make an interesting connection of a subroutine in our uniform rate mechanism TM-UNIFORM to the now well studied problem of online bipartite-matching and adwords allocation problem. We use this connection (in particular a result from [9]) to design a randomized mechanism TM-RANDOMIZED that attains an approximation factor of $\frac{2e-1}{e-1} \approx 2.58$. We further extend our mechanisms for many-to-many matchings in Section 6 (where each task needs to be done several times and each worker can do multiple tasks; as well as when utility of doing a task multiple times is given by a non-decreasing concave function). Finally, we carry out extensive experimentation on a realistic case study of Wikipedia translation project using Mechanical Turk workers. We also do simulations on synthetic data to evaluate the performance on various parameters of the problem. Our results demonstrate the practical applicability of our mechanism.

## 2 Related Work

From a technical perspective, the most similar work to that of ours is that of budget-feasible mechanism design which was initiated in [4]. Subsequent research in this direction [7, 10, 5, 6] 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 (this is the 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 standard knapsack problem, the best known approxmation factor is $2 + \sqrt{2}$ (and $3$ for randomized) and for submodular function it is $8.34$ (and $7.91$ for randomized), given by [7]. We note that our model lies in between these two models. However, we make explicit use of the mathematical structure of matchings in bipartite graphs and the assumption of *large markets*([6]) to design mechanisms with much better approximation factors 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 [11, 8, 12]. Motivated from crowdsourcing settings, budget-limited multi-armed bandits have also been studied [13, 14, 15, 16]. 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 [17]. Another recent work [18] 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. In [19], they present a model of workers to estimate wages for the workers. In [20], they show an automated way to negotiate payments with workers on Mturk.

We would like to point that some of the recent advancements [21, 9, 22, 23, 24] in the theory of online algorithms for matching and allocation problems inspired from online advertising is also relevant for the crowdsourcing setting. In fact, we use one of the technical result of [9] in our randomized mechanism to improve the approximation factor.

# 3 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 the person $p$. Also, let $u_t$ denote the utility of a task $t \in T$. An edge $e = (p, t)$ in the graph represents the notion 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[1] which represents the winning people and the tasks that are allocated to them. In addition, the mechanism comes up with a payment for each winning person. The two main 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 shoudn't exceed the budget $B$. The mechanism has to achieve above two properties while trying to maximize the total utility obtained from the tasks that gets allocated.

## 3.1 Large Markets

Crowd-sourcing systems are excellent examples of *large markets*, *i.e.* the number of participants are large enough that no single person can affect the market outcome significantly. Formally speaking, the definition of *large market assumption* that we use in our results is the following: 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 $\frac{u_{\mathsf{max}}}{U^*}$ 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.

## 3.2 Definitions

Let $N(p)$ and $N(t)$ respectively denote the set of neighbors of a person $p$ and a task $t$ in th graph $G$. Also, for simplicity in notation, we sometimes denote $E(G)$ by $E$. For a matching $\mathsf{M}$ and a person $p \in P$, the match of $p$ in $\mathsf{M}$ is denoted by $\mathsf{M}(p)$ (possibly equal to $\emptyset$). *Cost* of $\mathsf{M}$, denoted by $c(\mathsf{M})$ is defined by $\sum_{(p,t) \in \mathsf{M}} c_p$. Also, *utility* of $\mathsf{M}$, denoted by $u(\mathsf{M})$ is defined by $\sum_{(p,t) \in \mathsf{M}} u_t$. For any two matchings $\mathsf{M}, \mathsf{N}$, let $\mathsf{M} \triangle \mathsf{N}$ denote the graph which contains only the edges that appear in exactly one of the matchings $\mathsf{M}, \mathsf{N}$. It is a well-known fact that such a graph is always a union of 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 $\alpha$ if the utility obtained by this mechanism is always at least $\frac{1}{\alpha}$ of the utility obtained by the this optimum.

# 4 The Uniform Mechanism (TM-UNIFORM)

In this section, we present a simple mechanism that we call the Uniform Mechanism. It is called the Uniform Mechanism due to its uniform payment rule: 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 in Section 4.2.

[1] we relax this constraint later in section 6 to include the case when a person can do multiple tasks and each task can be done multiple times

The Uniform Mechanism, although not being truthful, satisfies truthfulness in a weaker form, which we call *one-way-truthfulness*. Briefly, by this property, players only have incentive to report costs lower than their true cost. This property also comes handy in analyzing the performance ratio of the mechanism, showing that it achieves an approximation of 3 compared to the optimum solution if the costs of the workers were known to the requester. First we formally define the notion of one-way-truthfulness in Section 4.1. Then we discuss and analyze the mechanism in Sections 4.2, 4.3. In Section 4.4, we make the same mechanism truthful only by changing its payment rule and using a non-uniform payment rule.

### 4.1 Oneway-truthfulness

Think of 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 one-way-truthful mechanism, no seller wants to report a cost higher than its true cost regardless of what others do. Formally, a one-way-truthful mechanism is defined as follows:

**Definition** A mechanism $\mathcal{M}$ is one-way-truthful if for any seller $p \in P$ and $x > c_p$ we have

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

where $d_{-p}$ denotes any cost vector corresponding to the rest of players except $p$ and $u(x, d_{-p})$ denotes the utility of $p$ when he reports $x$.

### 4.2 Outline of the Mechanism

We give a rough description of the Uniform Mechanism before presenting it formally. The key concept in the mechanism is a *buck per bang* rate $r$. This rate represents the amount of money that the mechanism is willing to pay per unit of utility, e.g. $r = 2$ means each unit of utility worths 2 (dollars) to us. We sometimes call the *buck per bang* rate simply the *rate*, when it is clear from the context. 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 the copy of graph $G$ which only contains edges with rate at most $r$.

The mechanism starts with $r = \infty$ and it gradually decreases the rate $r$. For any fixed $r$, it constructs the graph $G(r)$. Then, the nodes in $P$ are visited one by one in the order of appearance in the permutation $\sigma$, which is a fixed permutation of the vertices of $P$. When a person $p$ is visited, the mechanism assigns $p$ a task which has the highest utility among all the tasks that can be currently assigned to $p$. Let $\mathsf{M}$ denote the matching produced after visiting all the nodes in $P$. If $r \cdot u(\mathsf{M}) > B$, then we decrease the rate $r$ slightly and repeat this procedure for the new $r$; otherwise, we stop. The payments are uniform, i.e. we pay $r \cdot u_{\mathsf{M}(p)}$ to person $p$.

### 4.3 Formal Description of the Mechanism

Before presenting the Uniform Mechanism formally, we need to define a sorted list of the edges of $G$ as follows: 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 in decreasing order, i.e. for $e_i, e_j$ 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 infinity. Then, the Uniform Mechanism is formally defined as Mechanism 1.

---

**Procedure** FindMatching

---

**input** : Graph $G(P, T)$, Permutation $\sigma$
**output**: A matching in $G$

$\mathsf{M} \leftarrow \emptyset$;
**for** $i \leftarrow 1$ **to** $|P|$ **do**
  Find the task $t$ with the highest utility which is available for $\sigma(i)$;
  $\mathsf{M} \leftarrow \mathsf{M} \cup (\sigma(i), t)$;
**end**
Return the matching $\mathsf{M}$;

---

To give more intuition on what Mechanism 1 does, recall the rough description of the mechanism in Section 4.2. There, we defined a buck per bang rate $r$, which was continuously decreasing until the mechanism stops. Here, we discuss this description in more details; it helps to understand Mechanism 1 better as it is more intuitive, also, it comes handy in the analysis of the mechanism.

**Mechanism 1:** UniformMechanism

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

```
G' = G;
for i ← 1 to m do
    M = FindMatching(G', σ);
    if bb(e_i) · u(M) ≤ B then
        r ← min( B/u(M), bb(e_{i-1}) );
        break;
    end
    E(G') ← E(G') − {e_i};
end
Return M as the final matching;
Make the uniform payments with rate r.
```

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 more than $r$ fall to the left of the line. In case of ties (in buck per bangs), 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. Stop when the matching produced by FindMatching$(G', \sigma)$ 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 we use in the analysis.

**Definition** In running Mechanism 1, we say the mechanism is at *time* $(r, e)$ if the last edge that has been removed from $G'$ is $e$ and the current rate (position of the sweep line) is $r$.

Now, we state our main results for Mechanism 1, namely, individual rationality, one-way-truthfulness, and efficiency. The proofs are given in Appendix A.

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

**Lemma 2.** *If a person $p$ reports a cost higher than $c_p$, then Uniform Mechanism assigns him either the same task (as if he has reported $c_p$) or no task at all.*

**Lemma 3.** *The uniform mechanism is one-way-truthful.*

**Theorem 1.** *The uniform mechanism is individually rational, one-way-truthful, and has approximation ratio* 3*.*

### 4.4 The Truthful Mechanism

We can modify TM-UNIFORM and make it truthful by changing the uniform payment rule. The allocation rule (selecting the matching) stays identical to Mechanism 1. In this section, we describe a payment rule which, along with the allocation rule in Mechanism 1, gives a truthful mechanism. The payment rule is simple and is inspired from [25].

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

Next, we state the results for truthfulness and budget feasibility, i.e. the payments will not exceed the budget. The proofs are given in Appendix A.

**Lemma 4.** *Mechanism 1 is budget feasible with the Non-uniform Payment Rule.*

**Theorem 2.** *Mechanism 1 is truthful with the Non-uniform Payment Rule.*

## 5 The Randomized Mechanism (TM-RANDOMIZED)

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

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

We briefly discuss the properties of the mechanism here and leave the details in Appendix B. Formal definitions of these properties appear in Appendix B.1.1. TM-RANDOMIZED produces a fractional matching (see Appendix B.1.2 for a definition); if a fractional matching is not acceptable as the outcome of the mechanism, e.g. tasks are not splittable, then we provide a way to convert (round) the produced fractional matching to an integral matching. The resulting mechanism, now producing an integral matching, remains individually rational, almost truthful, and also, will have the same approximation ratio. We describe the mechanism TM-RANDOMIZED in Appendix B.2 and Appendix B.3. More details on our rounding method appear in Appendix B.4. Theorem 3 states the main result for TM-RANDOMIZED, proof of which is given in Appendix B.

**Theorem 3.** TM-RANDOMIZED *is individually rational, truthful in large markets, and has approximation ratio* $\frac{2e-1}{e-1}$.

# 6 Extensions

We 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 Section 7.

- **Tasks can be done multiple times.** Consider the more general case when tasks can be done multiple times, with 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 create $d$ copies of his node and treat them as different individuals i.e. each copy appears once in the permutation $\sigma$.

All the properties that we proved for our mechanisms also hold in these extensions. In this section, we briefly verify this fact for our simpler (non-randomized) mechanisms; we refer the reader to the full version of the paper for seeing that the properties of TM-RANDOMIZED also hold in these extensions. 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 is directly implied from Lemma 2. 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 $\theta_i$ is defined as follows: If copy $p_i$ is assigned to no task by the mechanism, then $\theta_i = 0$, otherwise, $\theta_i$ is the highest cost that $p$ could report such that $p_i$ remains assigned to some task by the mechanism. Lemma 5 states the main result on the truthfulness, the proof is given in Appendix C.

**Lemma 5.** *Extension of Mechanism 1 for many-to-many assignments is truthful under the non-uniform payment rule.*

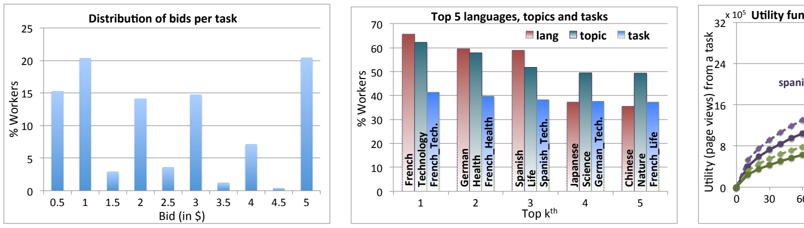
# 7 Experimental Evaluation

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 Mechanical Turk platform. We begin by describing our experimental setup, benchmarks and metrics.

## 7.1 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. This is a two factor approximation of the OPT, the untruthful optimal solution ([9]).
- UNTM-RANDOM is a trivial untruthful mechanism for matching which picks the edges (an available worker-task pair) in random order iteratively, paying the exact cost to the worker.




(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; along with *Culture* and *Life* as topics of interest. This corresponds to total of four 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*

- 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, else 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.

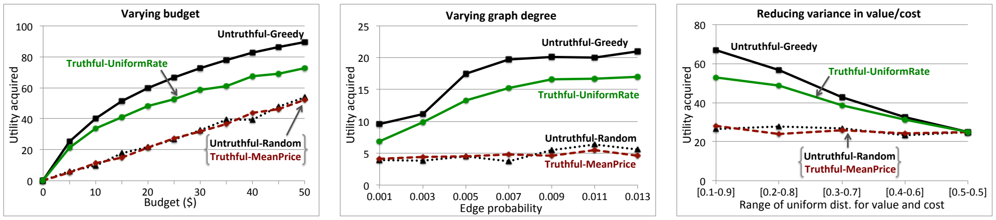
**Metrics and experiments.** The primary metric we track is the utility of the mechanism for a given budget. On synthetic data, we vary the amount of available budget to see the effect on our mechanism. We also vary degree of graph connectivity between tasks and workers to understand the effect of matching constraints. Additionally, we vary the variance of tasks' utilities and workers' cost to understand its impact, specially on the truthful mechanisms. In our experiments on MTurk data, utility directly maps 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 overall utility, we further track the utility acquired per target language and source topic, as would be explained further below. Our main goal is to gain insights in the execution of the mechanisms which arise from the market dynamics (for *e.g.* availability or shortage of specific skills and the utility difference between different type of tasks).

**Distributions and parameter choices.** For synthetic experiments, we considered 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 experiments on Wikipedia translation case study, the data was collected from Wikipedia, MTurk and other online resources. We describe the details of gathering real data for our experiments in Appendix D.1.

### 7.2 Results

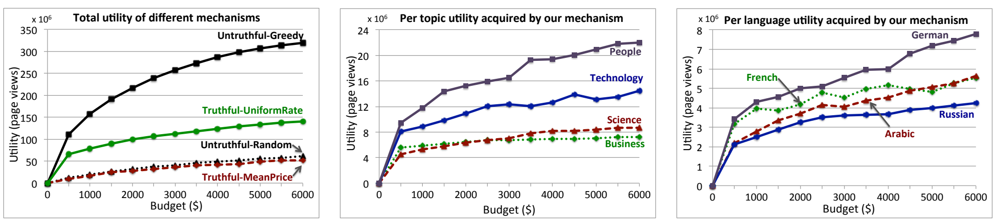
We now present our results on the data gathered as part of Wikipedia translation project using MTurk workers. We defer the discussion of results on synthetic data to Appendix D.2.

**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 effecting 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.




(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.




(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. 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*)

**Utility acquired per topic.** Next, we study the effect of market dynamics in 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 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, lets 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 higher than that for *Business* (49.50% compared to 34.72%).

**Utility acquired per language.** Along the same lines as above, Figure 3(c) illustrate 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 affects the availability of worker pool which in our data corresponds to 59.6%, 65.7%, 27% and 34.7% respectively.

## 8 Conclusions and Future Work

In this paper, inspired by the realistic crowdsourcing project of translating Wikipedia articles, we studied the mechanism design problem for crowdsourcing markets with matching like constraints. We give mechanisms with strong theoretical guarantees which are complemented by extensive experimentation to show the real-world applicability of these mechanisms. In future work, we think one should look at the case where workers can have different costs for different tasks. Another interesting generalization would be when tasks require multiple workers for them to be finished.

## References


[1] Mechanical turk platform. https://www.mturk.com/.

[2] Clickworker, virtual workforce. http://www.clickworker.com.

[3] Crowdflower. http://www.crowdflower.com/.

[4] Yaron Singer. Budget feasible mechanisms. In *FOCS*, pages 765–774, 2010.

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

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

[7] Ning Chen, Nick Gravin, and Pinyan Lu. On the approximability of budget feasible mechanisms. In *SODA*, 2011.

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

[9] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, 2008.

[10] Xiaohui Bei, Ning Chen, Nick Gravin, and Pinyan Lu. Budget feasible mechanism design: from prior-free to bayesian. In *STOC*, 2012.

[11] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: Posted price mechanisms for online procurement. In *EC*, 2012.

[12] Yaron Singer and Manas Mittal. Pricing mechanisms for crowdsourcing markets. In *WWW*, pages 1157–1166, 2013.

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

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

[15] Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *AAAI*, 2012.

[16] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R. Jennings. Efficient crowdsourcing of unknown experts using multi-armed bandits. In *AAAI*, 2012.

[17] Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, 2012.

[18] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Pick-a-crowd: tell me what you like, and i'll tell you what to do. In *WWW*, WWW '13, pages 367–374, 2013.

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

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

[21] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, 1990.

[22] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *SODA*, 2011.

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

[24] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *SODA*, 2013.

[25] Yaron Singer. Budget feasible mechanisms. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 765–774, Washington, DC, USA, 2010. IEEE Computer Society.

## A The Uniform Mechanism (TM-UNIFORM)

**Proof of Lemma 1.** Individual rationality follows from the fact that the buck per bang of each edge in $G^*$ is at most $r^*$. More precisely, for each edge $(p,t) \in G^*$ we have that $c_p/u_t \leq r^*$. So, if $p$ is assigned a task by the mechanism, then $c_p \leq r \cdot u_{\mathsf{M}(p)}$, which implies individual rationality. □

**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 stopping time of the mechanism 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.

If $r^* \leq \overline{r^*}$, then let $H, \overline{H}$ respectively denote $G'$ at time $(\overline{r^*}, \overline{e})$ in the real and fake instance. Also, denote the matchings produced by Running FindMatching on $H, \overline{H}$ respectively by $\mathsf{M}, \overline{\mathsf{M}}$.

We will prove that either $\overline{\mathsf{M}} = \mathsf{M}$ or $\overline{\mathsf{M}}(p) = \emptyset$. Let $t_1, \ldots, t_k$ be the set of tasks to which $p$ is interested in $H$, such that $u_{t_1} \geq \ldots \geq u_{t_k}$. For simplicity in 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 $p$. Also, let $\mathsf{M}(p)$ be denoted by $t_i$.

Since $p$ has reported a higher cost, a subset of its neighbors in $H$ become unavailable in $\overline{H}$. This subset must be of the form $\{t_j, \ldots, t_k\}$. If $j \leq i$, then we have $\overline{\mathsf{M}}(p) = \emptyset$, cause 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 $\overline{\mathsf{M}} = \mathsf{M}$.

It remains to show that the second case, $\overline{r^*} < r^*$, never happens. Let $H, \overline{H}$ respectively denote $G'$ at time $(r^*, e)$ in the real and fake instance. Also, let $\mathsf{M}, \overline{\mathsf{M}}$ respectively be the matchings produced by Running FindMatching on $H, \overline{H}$. We will prove that $u(\overline{\mathsf{M}}) \leq u(\mathsf{M})$ which is a contradiction: it implies that the stopping rate in the fake instance must have been at least $r^*$.

To this end, again let $t_1, \ldots, t_k$ be the set of tasks to which $p$ is interested in $H$, such that $u_{t_1} \geq \ldots \geq u_{t_k}$. If $p$ is assigned to a (non-empty) task $t_i$ in $\overline{\mathsf{M}}$, then it must be assigned to the same task in $\mathsf{M}$: it will be the highest available task that FindMatching can assign to it. Since $p$ is assigned to the same task in $\mathsf{M}, \overline{\mathsf{M}}$, then we have $\mathsf{M} = \overline{\mathsf{M}}$. So, assume that $p$ 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. This means $u(Q \cap \overline{\mathsf{M}}) \leq u(Q \cap \mathsf{M})$, which implies $u(\overline{\mathsf{M}}) \leq u(\mathsf{M})$.

□

**Proof of Lemma 3.** The proof is directly implied from Lemma 2. □

**Proof of Theorem 1.** The proof has two cases: we either spend the whole budget, or not. First we give the proof for the case when we spend the whole budget. 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 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 $U^+ \leq B/r$.

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.

Now we prove the case when we do not spend the whole budget. By showing that the left over budget is very small, we can follow the proof for the previous case. In fact, if we show that the left over budget is at most $ru_{\mathsf{max}}$, then instead of $U^+ \leq U$, we get $U^+ \leq U + u_{\mathsf{max}}$, 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 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 $\mathsf{M}$ which is budget feasible with the same rate $r$.

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 easy to verify that adding a single edge, $e$, to $H$, can not increase the utility of the output of FindMatching by more than $u_{\mathsf{max}}$. So, we have $u(\overline{\mathsf{M}}) \leq u(\mathsf{M}) + u_{\mathsf{max}}$. This fact, and the fact that $ru(\mathsf{M}) < B < ru(\overline{\mathsf{M}})$ imply that the left over budget is at most $ru_{\mathsf{max}}$. □

**Proof of Lemma 4.** Let $\mathsf{M}$ denote the matching produced in the last iteration of the mechanism. We show that the payment to person $p$ is not higher than $r^* u(\mathsf{M}(p))$. This will imply budget feasibility of the non-uniform payment rule due to budget feasibility of 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 remain a winner. Call the instances in which $p$ reports costs $c_p$ and $\overline{c_p}$ respectively the small and large instances. Let $(r^*, e)$ and $(\overline{r^*}, \overline{e})$ respectively denote the times at which the mechanism stops in the small and large instance. We will show that if $(r^*, e) \neq (\overline{r^*}, \overline{e})$ then we get contradiction. In the other hand, having $(r^*, e) = (\overline{r^*}, \overline{e})$ implies $\overline{c_p} \leq r^* \cdot u(\mathsf{M}(p))$, which would be a contradiction again.

To this end, first verify that $r^* \leq \overline{r^*}$, because otherwise, the mechanism finds a feasible matching in the large instance at rate $r^*$ as long as $\overline{c_p} > c_p$. This is simply due to the fact that it indeed found a feasible matching at rate $r^*$ in the small instance.

So, assume $r^* \leq \overline{r^*}$. Now, let $\mathsf{N}, \overline{\mathsf{N}}$ denote the matchings that the mechanism produces at time $(\overline{r^*}, \overline{e})$ 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 time $(\overline{r^*}, \overline{e})$ in the small instance.

While running the mechanism on the small instance, consider when it reaches to time $(\overline{r^*}, \overline{e})$. 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 time $(\overline{r^*}, \overline{e})$, 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. This is a contradiction. □

**Proof of Theorem 2.** 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 $\overline{c_p} \neq c_p$. Since $p$ can not win if it reports $c_p$ (or higher), then the payment to $p$ is bounded by $c_p$, which means, even if it misreports and wins, its utility will be zero. So, $p$ has no incentive to lie. □

# B The Randomized Mechanism (TM-Randomized)

## B.1 Preliminaries

### B.1.1 Truthful Mechanisms in Large Markets

In our setting, we say that a mechanism is *truthful in large markets* if, as the market becomes larger, i.e. $\frac{u_{\mathsf{max}}}{U^*}$ tends to zero, the following holds for any $p \in P$:

1. $\left(\sup_{\overline{c_p}} u(\overline{c_p}, d_{-p}) - u(c_p, d_{-p})\right) \to 0$
2. $c_p/\overline{c} \to 1$
3. $c_p/\underline{c} \to 1$

where denotes $d_{-p}$ any cost vector corresponding to the rest of players except $p$, $u(x, d_{-p})$ denotes the utility of player $p$ when he reports a cost $x$, and also

$$\overline{c} = \sup\left\{\overline{c_p} : u(\overline{c_p}, d_{-p}) > u(c_p, d_{-p})\right\}$$

and

$$\underline{c} = \inf\left\{\overline{c_p} : u(\overline{c_p}, d_{-p}) > u(c_p, d_{-p})\right\}.$$

In simple words, Condition 1 says: The extra utility that a person gains by misreporting his cost goes to zero, and Conditions 2,3 together say: Ratio of any beneficial misreported cost to the true cost goes to one.

#### B.1.2 Matchings

A fractional matching $x$ is a vector that has an entry $x_e$ for each edge $e$ in $G$ and moreover, it satisfies the following conditions

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

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

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 the mechanism is a special fractional matching that we define in this section. For any graph $G(P, T)$ and permutation $\sigma$ on the nodes of $P$, let $x(G, \sigma)$ denote the matching that is constructed by Procedure FindMatching$(G, \sigma)$. Then, 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 $P$ and $x(G, \sigma) \in \mathbb{R}^m$ is the characteristic vector of the integral matching that is produced from permutation $\sigma$.

Although we can not compute $x(G)$ in polynomial time, by sampling (polynomially) many permutations, it can be computed with arbitrary small error.

### B.2 Outline of the Mechanism

The mechanism roughly 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 we slightly decrease the rate $r$. This is done until we reach to the point that $r \cdot u(x(G(r))) \le B$.

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 $t$. We show that this mechanism is individually rational, truthful in large markets, and also, has approximation ratio $\frac{2e-1}{e-1}$.

**Mechanism 2:** 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$(G', \sigma)$;
  **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;
  **end**
  $E(G') \leftarrow E(G') - \{e_i\}$;
**end**
Make the uniform payments;
Return $x$ as the final matching;

### B.3 Formal Description of the Mechanism

Recall the sorted list of the edges of $G$, $e_1, \ldots, e_m$, in which the edges are sorted w.r.t. their buck per bang in decreasing order. Given this list, TM-RANDOMIZED is formally defined by Mechanism 2.

Mechanism 2 is individually rational, truthful in large markets, and has approximation ratio $\frac{2e-1}{e-1}$. We defer the proofs for individual rationality and truthfulness to the full version of the paper. Here, we only state the proof for the approximation ratio.

**Proof of Theorem 3.** We only prove the approximation ratio here. The idea is that although Procedure FindMatching is only a 2-approximation for a fixed permutation $\sigma$, it becomes a $\frac{e}{e-1}$-approximation if permutation $\sigma$ is chosen uniformly at random. This holds due to Section 3 of [9].

The proof follows the argument used for 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 which is gained 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 [9]: Procedure FindMatching is a $\frac{e}{e-1}$-approximation if permutation $\sigma$ is chosen uniformly at random. In the other hand, the expected utility of Procedure FindMatching when $\sigma$ is chosen randomly, is equal to utility of the fractional matching produced by taking the average of the outputs of FindMatching over all permutations $\sigma$. □

### B.4 Rounding

In this section, we give a high level description of the rounding procedure. Details of rounding are quite technical and involves 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.

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 where $x_i$ is chosen with probability $\lambda_i$.

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. All we need to do for getting a (strictly) budget feasible integral matching is running the mechanism with a slightly lowered budget. This can be done without any loss in the approximation ratio.

# C Extensions

**Proof of Lemma 5.** Fix a person $p$ and suppose he is willing to do up to $d$ tasks, which means he has $d$ copies in the graph, namely $p_1, \ldots, p_d$. We prove that for any copy $p_i$, person $p$ can never get a utility higher than $\max\{0, \theta_i - c_p\}$ from that copy, which implies truthfulness of the mechanism (due to its individual rationality).

First, verify that if $p_i$ is assigned to some task, then by definition, $\theta_i$ is the highest payment that $p$ can ever get for copy $p_i$; i.e. he will not get paid more for $p_i$ by misreporting. This means he can not increase his utility from copy $p_i$ by misreporting.

In the other hand, if $p_i$ is assigned to no tasks, then by the definition of the payment rule, it means $p$ will never get paid more than $c_p$ for copy $p_i$ (even by misreporting). So, his utility from $p_i$ will never be more than $0$. □

# D Experimental Evaluation

## D.1 Wikipedia translation on Mechanical Turk

We now describe our Wikipedia translation project in detail including data collection from online resources and worker's preference elicitation from Mechanical Turk 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 on 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 [2]. Next we considered the 10 most widely used internet languages (after English), along with their user base on internet [3] [4]. This 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 [5] along with their page view count. Next, we 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, we refer to it as *sub-tasks*, which equals the number of pages annotated with the topic of this task. 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 sub-task). These utilities for all the sub-tasks (ordered in their decreasing value) of a task form the concave utility curve associated with the task (Section 6). This is illustrated by an example in Figure 1(c).

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

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

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

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

**MTurk data and worker's preferences.** Next, we wanted to infer worker preferences in terms of topical interests as well the target languages they are interested in. We posted a HIT on Mechanical Turk 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 be potentially trained with set of tools to assist in our translation project. Our survey explicitly asked following questions to 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 in owning
- 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 own 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 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 from workers. 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 sub-tasks associated with these tasks.

### D.2 Results on Synthetic data

**Varying budget.** Figure 2(a) shows the results of 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. e.g. 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 shall 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, effecting the performance of the mechanisms. Figure 2(b) study 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 same. And, as variance in market increases, the relative performance of our mechanism TM-MEANPRICE *w.r.t* UNTM-GREEDY decreases. Intuitively, this means that markets with higher variance increases the strategic power of the workers.