3M4-OS-07e-3

クラウドソーシングにおける複数タスク割当て

Multi-unit Task Allocation in Crowdsourcing

松原繁夫[*1] 水島拓也[*2]
Shigeo Matsubara Takuya Mizushima

[*1]京都大学 社会情報学専攻 Kyoto University
[*2]京都大学工学部情報学科 Kyoto University

We propose a mechanism for allocating multi-unit tasks to workers that elicits truthful preferences for completing tasks. Truthful reporting can be achieved if at most one task is allocated to each worker or if monetary transfer is allowed. However, it is not sufficiently clear when multi-unit tasks might be allocated to a worker and monetary transfer is not allowed for determining the allocation. We assume that mechanism designer has prior knowledge of preference distribution, i.e., which tasks are preferred by many workers and which ones are preferred by few workers, but does not know the true preference of an individual worker. First, we characterize workers' strategic reporting and discuss the reduction in social surplus, i.e., the sum of a worker's utilities. Second, we propose a mechanism for preventing strategic reporting by introducing probabilistic allocation skipping and clarifying the condition of skip probability to enable truthful reporting of all workers. Experimental results show in which situation the proposed mechanism can improve allocation efficiency compared to a simple allocation mechanism.

1. はじめに

近年，Amazon Mechanical Turk (MTruk) の利用増などに伴い，人工知能技術の適用対象としてクラウドソーシングが注目を集めている．本稿ではクラウドソーシング環境におけるワーカへのタスク割当てについて議論する．クラウドソーシングでタスクを処理する場合の経験則として，複雑なタスクであれば，それを単純なサブタスクに分割することが勧められている．例えば，文書校正タスクであれば，誤りを発見するタスク，改訂案を提案するタスク，複数の改訂案の中から最善のものを選択するタスクに分割することで，タスク処理品質の改善が見込めるとされている．ここで，各サブタスクに対するワーカの能力が異なるとすれば，高品質な結果を得るためには，どのタスクをどのワーカに割り当てるを考えねばならない．

MTurk のワーカは，新着順に閲覧してタスク選択することが多いと報告されている [Ipeirotis 10a]．しかし，この観点からリクエスタがタスク割当てを制御しようとしても，最適なタスク依頼時期を判断することは難しい．ワーカの目に留まり易くするように同じタスクを何度も依頼する方法もあるが，これは費用増を招く．

望ましいタスク割当てを実現する別の方法は報酬の制御である．ワーカを集めにくいタスクに関しては報酬を上げ，集めやすいタスクに関しては報酬を下げることで，ワーカを分散させようとする考えである．しかし，この単純な方法はうまく機能しないと考えられる．それは，クラウドソーシング市場には多くの不誠実なワーカが存在するからである [Ipeirotis 10b]．報酬を高額にすれば，不誠実なワーカを集めることになり，却って品質が低下するかもしれない．もちろん，タスクを完了しなかったワーカには報酬を支払わないこともできるが，リクエスタとワーカの相互選択というクラウドソーシング市場の性質上，不誠実なワーカを十分に排除することは難しい．

ここで，リクエスタが種類の異なる複数のタスクを持つとき，報酬額調整に頼らずに，どのように望ましい割当てを実現するかという問題が生じる．この問題は組合せ最適化問題の一つである割当問題として定式化できる．タスク集合とワーカ集合があるときに，制約を満たすようにタスクとワーカの組を求める問題であり，クラス割当問題など多くの応用が存在する．効率的なタスク処理を実現するには，個々のワーカの能力に応じてタスクを割り当てることが必要である．しかし，リクエスタにとって個々のワーカの能力を直接観測することは難しく，ここに，インセンティブの問題が生じる．

もし単純に整数割当問題として定式化して割当てを求めれば，ワーカは戦略的に振る舞うことで，自己の効用を増加させ得る．このような戦略的な振る舞いを防ぐ方法の 1 つとして，耐戦略性を持つ Vickrey-Clarke-Groves (VCG) メカニズムが存在する．しかし，先に述べたように高額報酬は不誠実なワーカを引き寄せ，必ずしも効率的なタスク処理を保証しない．また，耐戦略性を持つ別のメカニズムとして，Random Serial Dictatorship (RSD) メカニズムが存在する [Abdulkadiroğlu 98]．しかし，このメカニズムは耐戦略性を満たすものの，非効率な割当てが得られる場合もあり，また，公平性という点で問題がある．

この問題を解決するため，本稿では新たなタスク割当メカニズムを提案する．提案メカニズムでは，人気タスク，すなわち，多くのワーカが選択を希望するタスクの割当てを受けたワーカには，それ以外のタスクの割当てを確率的にスキップすることで，ワーカの虚偽申告を防ぐことを試みる．

以下 2 章で関連研究について述べ，3 章でタスク割当問題のモデルを提示する．4 章で既存手法の問題点を指摘し，5 章で新たな割当メカニズムを提案する．6 章で提案手法の評価を行い，7 章をむすびとする．

2. 関連研究

Random Serial Dictatorship (RSD) メカニズムはこれまで広く研究されている．Budish らは複数ユニット割当問題における誘因両立性の問題を研究している [Budish 12]．Che らは RSD メカニズムの派生メカニズムである確率的 Serial Dictatorship メカニズムを取り上げ，効率性の観点から RSD メカニズムと比較している．これらの研究と比べて，本研究ではメカニズム設計者が選好分布に関する事前知識を持つと仮

連絡先: 松原繁夫，京都大学，京都市左京区吉田本町

定するところが異なる．すなわち，個々のワーカの選好はわからないが，どのタスクが多くのワーカによって選好されるかを知っているとの仮定の元で，人気タスクに対するワーカの虚偽申告を防ぐことに焦点を当てている．

また，意思決定論を用いて，ワークフロー実行を動的に制御しようとする研究も存在する [Dai 10, Kern 10, Parameswaran 12]．しかし，それらの研究では誘因両立性の問題，すなわち，ワーカの戦略的行動が考慮されておらず，本研究はその点で異なる．

3. タスク割当問題

本章ではタスク割当問題を定式化する．複数タスクと複数ワーカが存在する状況を扱う．ワーカはどのタスクでも請け負えるが，タスク完了に要する費用が異なる．本稿の目的は，全タスクが完了されるという制約の元で，社会的余剰を最大化すること，すなわち，ワーカに発生する費用の総和を最小化することである．

タスクは $T = \{t_1, t_2, \cdots, t_m\}$ として表現される．各タスクは k 人のワーカに割り当てられる．これは，品質保証のために同じタスクを複数のワーカに割り当てる場合や，あるいは，同種のタスクが多数あり，完了時間を早めるために，複数ワーカに少しずつ分担させる場合などを想定している．なお，議論を簡単にするため，k の値は全てのタスクで同じと仮定する．

ワーカは $W = \{w_1, w_2, \cdots, w_n\}$ として表現される．各ワーカは異なる能力を持つとする．一旦各タスクの報酬額が与えられれば，各ワーカはタスクに関して選好順序を決定できる．なお，選好順序に同順位は存在しないと仮定する．l 番目に選好するタスクから得られる効用を v_l と表す．やや強い仮定であるが，l 番目に選好するタスクから得られる効用はワーカ間で等しいと　仮定する．ワーカ w_i がタスク t と t' の割当てを受けた場合，ワーカの効用はタスク t から得られる効用と t' から得られる効用の和として計算される．つまり，別のタスクの存在によって，あるタスクの処理費用が増加することも減少することもないと考える．

つぎに選好分布について議論する．まず，人気タスクを以下のように定義する．

定義 1 ワーカが *1* つタスクのみ選択することが許されるとき，希望数が割当数 k に達するならば，タスク t を人気タスクと呼ぶ．

人気があるということは，タスク割当希望に関して競合が生じることを意味する．本稿では，人気タスク／不人気タスクの 2 つのクラスのみを考え，$T_{popular}$ で人気タスクの集合を，$T_{unpopular}$ で不人気タスクの集合を表す．

リクエスタはワーカの選好分布を知っている，すなわち，どのタスクが人気タスクか，どのタスクが不人気タスクかを知っているが，個々のワーカの選好順序は知らないと仮定する．リクエスタの意思決定問題は，タスクの割当て方であるとし，本稿では報酬額の設定は与えられているものと考える．

4. 従来法の問題点

タスク割当問題を解くメカニズムの一つは Random Serial Dictatorship (RSD) メカニズムを用いることである．これはワーカの順序をランダムに決定し，順に残っているタスクの中から希望するタスクを割り当てていく方式である．4 つのタスクと 4 つのワーカが存在し，以下の選好を持つ場合を考える．

w1: t1 ≻ t4 ≻ t3 ≻ t2
w2: t1 ≻ t2 ≻ t3 ≻ t4
w3: t2 ≻ t1 ≻ t3 ≻ t4
w4: t3 ≻ t1 ≻ t2 ≻ t4

各タスクの割当上限を 1 と考える．また，$v_1 = 4$, $v_2 = 3$, $v_3 = 2$,$v_4 = 1$ を仮定する．もし $w1$, $w2$, $w3$, $w4$ の順でタスクを選択する場合，割当ては ($t1$, $t2$, $t3$, $t4$) = ($w1$, $w2$, $w3$, $w4$) となり，社会的余剰は 4+3+2+1=10 となる．ここで，($t1$, $t2$, $t3$, $t4$) = ($w2$, $w3$, $w4$, $w1$) と割り当てれば，社会的余剰 3+4+4+4 = 15 を実現できる．よって，RSD メカニズムは効率的でない割当てを生成する場合があることが確認できる．

別のメカニズムはワーカから選好リストを集めて，整数計画問題として解く方法である．ここで，社会的余剰を最大にする割当て方が複数存在すれば，各ワーカの効用ができるだけ等しくなる割当て方を選択する．この方式は RSD メカニズムと比べて社会的余剰を改善できる可能性がある．しかし，この方式は耐戦略性を持たない．上記の例において，$w1$ は以下のように虚偽の選好を申告することで，効用を増加させ得る．

w1: t1 ≻ t3 ≻ t2 ≻ t4
w2: t1 ≻ t2 ≻ t4 ≻ t3
w3: t2 ≻ t1 ≻ t3 ≻ t4
w4: t3 ≻ t1 ≻ t2 ≻ t4

真実申告の場合の割当て ($t1$, $t2$, $t3$, $t4$) = ($w2$, $w3$, $w4$, $w1$) に対して，虚偽申告の場合の割当ては ($t1$, $t2$, $t3$, $t4$) = ($w1$, $w3$, $w4$, $w2$) となり，ワーカ $w1$ は第 2 希望のタスクではなく第 1 希望のタスクを獲得できる．

ここまでは割当上限が 1 の例を見てきたが，つぎに 2 の場合を見てみる．5 つのタスクと 5 人のワーカが存在する以下の例を考えよう．

w1: t4 ≻ t1 ≻ t3 ≻ t5 ≻ t2
w2: t1 ≻ t2 ≻ t3 ≻ t4 ≻ t5
w3: t1 ≻ t5 ≻ t2 ≻ t4 ≻ t3
w4: t2 ≻ t4 ≻ t5 ≻ t3 ≻ t1
w5: t3 ≻ t5 ≻ t1 ≻ t2 ≻ t4

ワーカが真実申告すれば，社会的余剰を最大にする割当てとして ($t1$,$t2$,$t3$,$t4$,$t5$)=(($w2$,$w3$),($w2$,$w4$), ($w1$,$w5$), ($w1$,$w4$), ($w3$,$w5$)) が選択され，社会的余剰は 44 となる．しかし，この場合も真実申告が均衡戦略とはならない．ワーカ $w1$ は $t1$ に関して過大申告する誘因を持つ．$w1$ が以下を申告し，他のワーカが真実申告した場合を考えよう．

w1: t1 ≻ t4 ≻ t3 ≻ t5 ≻ t2

この場合，社会的余剰を最大にする割当ては ($t1$,$t2$,$t3$,$t4$,$t5$) = (($w1$,$w3$), ($w2$,$w4$), ($w2$,$w5$), ($w1$,$w4$), ($w3$,$w5$))，あるいは，(($w2$,$w3$), ($w2$,$w4$), ($w1$,$w5$), ($w1$,$w4$), ($w3$,$w5$)) となる．前者では，$w1$ の効用は 8 から 9 に増加するが，社会的余剰は 44 から 43 に減少する．虚偽申告による社会的余剰の低下は，ワーカが第 1 希望タスクからより大きな効用を得る場合に深刻となる，例えば $v_1 = 10$ といった場合，虚偽申告によって社会的余剰が 69 から 65 に減少する．

5. 提案方法

5.1 基本となる考え方

タスク $t1$ の割当数が $t1$ を第 1 希望とするワーカ数より小さい場合を考えよう．このとき，$t1$ を第 2 希望とするワーカが選好リストを真実申告すれば，$t1$ の割当てを受けることはない．しかし，$t1$ を第 1 希望と申告すれば，$t1$ の割当てを受

ける可能性が生じる．ここで，真の第 1 希望タスクが不人気タスクである場合，人気タスク $t1$ を第 1 希望としても，真の第 1 希望タスクの割当てを受けられなくなるリスクは非常に小さいと言える．

この問題を解決するため，人気タスクを割り当てるワーカには，その他のタスクの割当てを決める際に，確率的に割当てをスキップすることを考える．割当スキップ確率を p で表す．4 章での 5 人のワーカの例を考えると，ワーカ $w1$ が真実申告した場合，タスク $t3$ と $t4$ の割当てを受け，効用は 5+3=8 となる．一方，$w1$ がタスク $t1$ を第 1 希望と過大申告して $t1$ の割当てを受けることに成功した場合を考える．リクエスタにとっては，これが真実申告か虚偽申告か直接確認することはできないが，人気タスク $t1$ の割当てを受けたということで，つぎの $t4$ の割当てを確率 p でスキップする．この場合の効用は $4+5(1-p)$ となる．よって，$p > 0.2$ が成立すれば，ワーカ $w1$ にとって真実申告する方が大きな効用を得られることになる．こうして，ワーカ $w1$ は自発的に人気タスクへの過大申告を取り止めることになる．

確率的割当スキップメカニズム

1. リクエスタはワーカに選好リストを申告するよう依頼する（申告された選好リストは真の選好とは限らない）．
2. 申告された選好リストの元で，整数計画問題を解く．
3. 人気タスクの割当てを受けるワーカに対して，他のタスクの割当てを確率 p でスキップするように割当案を修正する．
4. 人気タスクの割当てを受けたワーカへの割当てを固定し，スキップされたタスクを含めて再度整数計画問題を解いて残りのタスクの割当てを決定する．

ここでの問題は確率 p をどのように設定するかである．$t1$ を人気タスクとして，p を大きな値に設定すれば，$t1$ を真の第 2 希望とするワーカが $t1$ を第 1 希望と申告することを防げる．しかし，$t1$ を第 1 希望とするワーカまでも，$t1$ を第 1 希望として申告しなくなる恐れがあり，社会的余剰の低下を招く．一方，p を小さな値に設定すれば，$t1$ を真の第 2 希望とするワーカが $t1$ を第 1 希望と過大申告することを防げなくなる．以下，適切な確率 p の求め方について述べる．

5.2 準備

仮定 1 *1* つの人気タスクが $t1$ が存在する．すなわち，$T_{popular}$ の要素は *1* つである．$t1$ を第 *1* 希望としないワーカに関しては，$t1$ に対するワーカの選好は第 *2* 希望から第 m 希望まで均等に分散している．また，不人気タスクについては，ワーカの選好は均等に分散している．

これらの仮定はかなり強いものであるが，以降の計算を簡潔に示すために，本稿では仮定する．人気タスク $t1$ を第 1 希望とするワーカの比率を α で表す．あるワーカにとって，人気タスク $t1$ の選好順位が l である場合，そのワーカのタイプが θ_l であると呼ぶ．

このとき，人気タスク $t1$ 第 1 希望ワーカは αn 人，$t1$ 第 2 希望ワーカから第 m 希望ワーカは各々 $(1-\alpha)n/(m-1)$ 人存在する．不人気タスク $t2$ 第 1 希望ワーカは $(1-\alpha)n/(m-1)$ 人，$t2$ 第 2 希望ワーカは，$t1$ 第 1 希望ワーカの中に $\alpha n-(1-\alpha)n/(m-1)$ 人，$t1$ 以外第 1 希望ワーカの中に $(1-\alpha)n/(m-1)$ 人いるため，$(\alpha n-(1-\alpha)n/(m-1))/(m-1)+(1-\alpha)n/(m-1) = (n/(m-1))(1-(1-\alpha)/(m-1))$ 人と表わされる．他の不人気タスク $t3, t4$ などについても，$t2$ と同様である．

5.3 単純な割当メカニズムによる獲得効用

単純な割当てメカニズムとは，(1) ワーカに選好リストを申告するよう依頼し，(2) 整数計画問題として，社会的余剰を最大にする割当てを求める方法である．紙幅の制限により，本稿では $k \leq \alpha n$ の場合について記述する．$k > \alpha n$ の場合についても同様な議論が可能である．

各ワーカは mk/n 個のタスクを割当てられる．この場合，タイプ θ_l $(2 \leq l \leq mk/n)$ のワーカは，真実申告した場合，人気タスク $t1$ の割当てを受けることができない．よって，効用は以下のように計算される．

$$u(\theta_l';\theta_l) = \sum_{i=1}^{mk/n+1} v_i - v_l \qquad (2 \leq l \leq mk/n)$$

ここで，$u(\theta_l';\theta_l)$ は真のタイプが θ_l であるワーカが θ_l' と申告した場合の効用を表す．

一方，タイプ θ_l $(2 \leq l \leq mk/n)$ のワーカが人気タスクに対して過大申告した場合，その効用は以下のように計算される．

$$u(\theta_1';\theta_l) = \frac{k}{\alpha n+1}\sum_{i=1}^{mk/n} v_i + (1-\frac{k}{\alpha n+1})(\sum_{i=1}^{mk/n+1} v_i - v_l)$$

右辺第 1 項は当該ワーカが人気タスクの割当てを受けた場合の効用を表し，右辺第 2 項は人気タスクの割当てを受けられなかった場合の効用を表す．つねに，$u(\theta_1';\theta_l) > u(\theta_l';\theta_l)$ が成立するため，タイプ θ_l $(2 \leq l \leq mk/n)$ のワーカは人気タスク $t1$ に対して過大申告するよう動機づけられる．

5.4 確率的割当スキップメカニズムにおける獲得効用

本節でも，紙幅の制限により，$k \leq \alpha n$ の場合についてのみ記述する．まず，タイプ θ_l $(2 \leq l \leq mk/n)$ のワーカについて考える．真実申告した場合の効用と，人気タスク $t1$ に対して過大申告した場合の効用は以下のように計算される．

$$u(\theta_l';\theta_l) = \sum_{i=1}^{mk/n+1} v_i - v_l + \frac{pk(mk/n-1)}{(1-\alpha)n} v_{\frac{mk}{n}+1}$$

$$u(\theta_1';\theta_l) = \frac{k}{\alpha n+1}(\sum_{i=1}^{mk/n} v_i - p(\sum_{i=1}^{mk/n} v_i - v_l)) + (1-\frac{k}{\alpha n+1})(\sum_{i=1}^{mk/n+1} v_i - v_l)$$

$u(\theta_l';\theta_l)$ に関して，右辺第 3 項は θ_1 ワーカに対して割当てスキップされたタスクの再割当てに対応する項である．人気タスクに対するワーカの過大申告を防ぐためには，$u(\theta_l';\theta_l) \geq u(\theta_1';\theta_l)$ が成立する必要がある．この条件より，確率 p の下限が求まる．

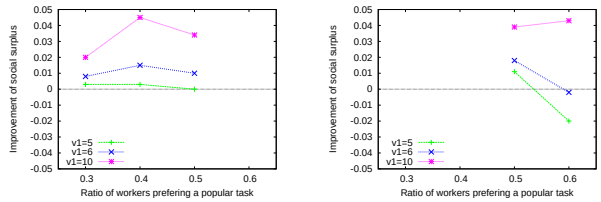
つぎに，タイプ θ_1 のワーカについて考える．真実申告した場合と人気タスクに対して過小申告した場合の効用は以下のように計算される．

$$u(\theta_1';\theta_1) = \frac{k}{\alpha n}(\sum_{i=1}^{mk/n} v_i - p\sum_{i=2}^{mk/n} v_i) + (1-\frac{k}{\alpha n})\sum_{i=2}^{mk/n+1} v_i$$

$$u(\theta_2';\theta_1) = \sum_{i=2}^{mk/n+1} v_i + \frac{pk(mk/n-1)}{(1-\alpha)n+1} v_{\frac{mk}{n}+1}$$

表 1: スキップ確率 p の範囲

(m,k,n)	α	$v_1=5$ $\underline{p}$	$v_1=5$ $\overline{p}$	$v_1=6$ $\underline{p}$	$v_1=6$ $\overline{p}$	$v_1=10$ $\underline{p}$	$v_1=10$ $\overline{p}$
(5,4,10)	0.3	0.09	0.32	0.07	0.48	0.04	1
	0.4	0.20	0.50	0.17	0.75	0.10	1
	0.5	0.20	0.50	0.17	0.75	0.10	1
(5,6,10)	0.5	0.05	0.24	0.04	0.32	0.03	0.64
	0.6	0.25	0.43	0.22	0.57	0.15	1
(10,6,20)	0.3	0.25	0.43	0.22	0.57	0.15	1
	0.4	0.25	0.43	0.22	0.57	0.15	1
	0.5	0.25	0.43	0.22	0.57	0.15	1
(10,8,20)	0.4	0.30	0.44	0.27	0.56	0.20	1
	0.5	0.30	0.44	0.27	0.56	0.20	1
	0.6	0.30	0.44	0.27	0.56	0.20	1



(a) (m,k,n)=(5,4,10)　(b) (m,k,n)=(5,6,10)

(c) (m,k,n)=(10,6,20)　(d) (m,k,n)=(10,8,20)

図 1: 社会的余剰の比較

人気タスクに対するワーカの過小申告を防ぐためには，$u(\theta_1';\theta_1) \geq u(\theta_2';\theta_1)$ が成立する必要がある．この条件より，確率 p に対する上限が求まる．

$k > \alpha n$ の場合についても同様に確率 p が満たす条件を求めることができ，以下の性質を証明することができる．

命題 1 仮定 *1* のもとで，確率的割当スキップメカニズムは人気タスクに対するワーカの過大申告・過小申告を防ぐ．

証明 1 簡単な計算により，上記の条件を満たす確率 p がつねに存在することが示される．

6. 評価実験

本章では，社会的余剰に関して，単純な割当メカニズムと提案方法を比較する．m, k, n, α, p といったパラメータ値を与えれば，5 章で示した効用計算により社会的余剰を計算できる．なお，単純な割当メカニズムにおいては，タイプ θ_l $(2 \leq l \leq mk/n)$ のワーカは，人気タスクに対してつねに過大申告すると仮定して社会的余剰を計算した．

実験設定を以下に示す．人気タスクは 1 つのみ存在すると仮定し，第 1 希望のタスク請負からの効用に関して $v_1 = 5, 6, 10$ の 3 つの場合について調べた．第 2 希望以下の効用は $v_2 = 4$, $v_3 = 3$, $v_4 = 2$, $v_5 = 1$ に固定した．また，$m = 10$ の場合，$v_i = 0$ (for $i \geq 6$) とした．α の値は $\lfloor mk/n \rfloor = mk/n$ という関係が充足されるように選択した．表 1 にスキップ確率 p を計算した結果を示す．例えば，$(m,k,n) = (5,4,10)$ の場合，$p = 0.2$ と設定すれば真実申告が実現される．

図 1 は社会的余剰の比較結果を表す．ここでは，スキップ確率 p は真実申告を実現する最小値に設定されている．プロットされた値は増加率を表している．例えば，0.01 であれば，提案方法を用いることで，社会的余剰が 1% 増加することを意味する．提案方法は，mk/n の値が小さく，v_1 の値が大きいときに有効であることが図からわかる．

7. むすび

本稿では，クラウドソーシング環境におけるタスク割当問題に対して，リクエスタがワーカの選好分布を知っているとの仮定のもとで，ワーカの虚偽申告を防ぐメカニズムを提案した．事前にワーカが選好リストを申告するといった方法は，Amazon Mechanical Turk とは異なる方式であり，提案方法をどのように実環境に組み込むかは今後の課題である，

謝辞

本研究は，日本学術振興会科学研究費基盤研究 (S) (24220002, 平成 24 年度〜28 年度) の補助を受けた．

参考文献

[Abdulkadiroğlu 98] Abdulkadiroğlu, A. and Sönmez, T.: Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems, *Econometrica*, Vol. 66, No. 3, pp. 689–701 (1998)

[Budish 12] Budish, E. and Cantillon, E.: The Multi-unit Assignment Problem: Theory and Evidence from Course Allocation at Harvard, *American Economic Review*, Vol. 102, No. 5, pp. 2237–2271 (2012)

[Dai 10] Dai, P., Mausam, , and Weld, D. S.: Decision-theoretic control of crowd-sourced workflows, in *In the 24th AAAI Conference on Artificial Intelligence (AAAI'10*, pp. 1168–1174 (2010)

[Ipeirotis 10a] Ipeirotis, P. G.: Analyzing the Amazon Mechanical Turk marketplace, *XRDS*, Vol. 17, No. 2, pp. 16–21 (2010)

[Ipeirotis 10b] Ipeirotis, P. G., Provost, F., and Wang, J.: Quality management on Amazon Mechanical Turk, in *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pp. 64–67 (2010)

[Kern 10] Kern, R., Thies, H., and Satzger, G.: Statistical Quality Control for Human-Based Electronic Services, in *Proceedings of 8th International Conference on Service-Oriented Computing (ICSOC 2010)*, pp. 243–257 (2010)

[Parameswaran 12] Parameswaran, A. G., Garcia-Molina, H., Park, H., Polyzotis, N., Ramesh, A., and Widom, J.: CrowdScreen: algorithms for filtering data with humans, in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 361–372 (2012)