Guided Crowdsourcing for collective work coordination in corporate environments

Ioanna Lykourentzou[1], Dimitrios J. Vergados[2], Katerina Papadaki[3], and Yannick Naudet[1]

[1] Centre de Recherche Public Henri Tudor, Luxembourg
ioanna.lykourentzou@tudor.lu, yannick.naudet@tudor.lu
[2] Norwegian University of Science and Technology, Department of Telematics, Norway
dimitrios.vergados@item.ntnu.no
[3] Bank of Greece, Operational Risk Management, Greece
kpapadaki@bankofgreece.gr

Abstract. Crowdsourcing is increasingly gaining attention as one of the most promising forms of large-scale dynamic collective work. However current crowdsourcing approaches do not offer guarantees often demanded by consumers, for example regarding minimum quality, maximum cost or job accomplishment time. The problem appears to have a greater impact in corporate environments because in this case the above-mentioned performance guarantees directly affect its viability against competition. Guided crowdsourcing can be an alternative to overcome these issues. Guided crowdsourcing refers to the use of Artificial Intelligence methods to coordinate workers in crowdsourcing settings, in order to ensure collective performance goals such as quality, cost or time. In this paper, we investigate its potential and examine it on an evaluation setting tailored for intra and inter-corporate environments.

Keywords: crowdsourcing; crowd coordination; resource allocation

1 Introduction

Crowdsourcing is a new form of user involvement on the Web. It has recently emerged as a new paradigm of collective work and as a natural result of the Web's evolution course, from a purely non-participatory system, with users in the place of content consumers, to a virtual space of full user involvement, since the Web 2.0 era and beyond.

Crowdsourcing refers to the splitting of a large, human-intelligence job into smaller micro-tasks and dynamically "outsourcing" these, not to specific individuals, but to an unknown crowd of web workers. Examples of jobs often accomplished through crowdsourcing include the translation of large corpuses of small sentences from one language to the other, the recognition of captchas, the transcription of audio files to text, but also the collective creation of articles in Wikipedia and the development of open source software artifacts by several distributed programmers [4].

The crowdsourcing technology increases rapidly. Having started only a few years ago, it is already being used at large-scale by commercial players, academics and individuals, who benefit from its ability to involve millions of users worldwide and to

provide access to a scalable and on-demand workforce. Indicative of its prospective, crowdsourcing was recently included to the cycle of emerging technologies with significant foreseen potential, as predicted by professional technology watch firms like Gartner[1].

Despite its success, crowdsourcing has often been criticized for not providing guarantees critical for the requesters, such as minimum job quality, maximum cost and timeliness [6]. This is because the participating workers select the micro-tasks that they will work on with an aim to maximize individual and not system-level targets. For example workers in paid crowdsourcing seek to increase their individual profit by focusing on quantity rather than quality (i.e. submitting more in number rather than high-quality tasks). This inability to guarantee performance, and to do so simultaneously for multiple performance objectives, hinders the reliability of crowdsourcing and limits its potential. Especially for corporate environments, the above limitations make the corporate management even more skeptical in incorporating crowdsourcing approaches in vital organizational processes. Thus, recent research has started to identify the need of standardizing crowdsourcing [7] and improving in terms of system-level performance, using artificial intelligence (AI).

In this paper we present this new area of guided crowdsourcing, which can be defined as using AI methods to coordinate a user crowd towards achieving specific collective performance goals in a crowdsourcing setting. In section 2 we formulate the engineering of guided crowdsourcing solution as a 5-step process. In section 3 we present the main research streams in the area. In section 4 we showcase its capabilities for a specific application case, i.e. corporate environments. Finally, in section 5 we discuss the open research topics related to engineering efficient guided crowdsourcing solutions and conclude the paper.

2 Guided Crowdsourcing: A new research area for crowdsourcing optimization through AI-based coordination

2.1 Definition and differences with standard crowdsourcing

Given the problems of current crowdsourcing, research has slowly started to consider what we may overall refer to as "Guided Crowdsourcing". Guided crowdsourcing can be briefly defined as *"the use of AI methods to coordinate and guide users participating in a crowdsourcing system towards achieving a collective result that meets specific performance standards, such as quality, timeliness or cost"*. The purpose of guided crowdsourcing is therefore to optimize the performance of the crowdsourcing system and provide quality, cost and time guarantees to the consumers.

Its difference with current, unguided crowdsourcing is that the latter is totally self-coordinated, with workers self-appointing themselves to the tasks that they wish to undertake, thus often resulting to poor performance results. In contrast, the coordination algorithms used in guided crowdsourcing are designed to affect the behavior of the

[1] Gartner's 2012 Hype Cycle for Emerging Technologies. Press Release. `http://www.gartner.com/it/page.jsp?id=2124315`.

workers towards a specific direction, in order to achieve a specific crowdsourcing performance result.

Affecting user behavior can be implicit (increasing the price of certain tasks to make users prefer them over others) or explicit (recommending tasks to specific users).

2.2 Guided crowdsourcing as a 5-step process

The basic elements that need to be defined to engineer a crowdsourcing solution out of a standard, unguided crowdsourcing system, are the following (2.2):

Goals. As goals, we define the performance aspects of the standard crowdsourcing system, which the guided approach targets at improving. They can include, as a non-exhaustive list, maximizing the quality of the accomplished jobs, minimizing the cost for each job, and meeting the deadlines of each job.

Jobs. For each crowdsourced job, we need to define one or more performance characteristics, based on the system's performance goals. These can include the jobs current quality level, current cost, deadline, as well as other more specific traits. The measurement of each characteristic of the job is either global, like its deadline, or an aggregation of the characteristics of the jobs micro-tasks. For example, if we assume a job of translating a corpus of sentences, the jobs current quality is the sum of the quality of the translation of each of the sentences (tasks) that have already been translated.

Workers. The workers participating in the standard crowdsourcing system are in fact its resources. For each worker, specific skills should be defined in relation to the system's goals. Such skills can for instance include the worker's expertise in relation to a knowledge-intensive task, task fulfillment speed, accuracy, judgment ability, as well as the minimum wage he would require to accomplish a given micro-task. The estimation of worker skills can be addressed through learning mechanisms, for example neural networks as used in [11].

Constraints. The constraints are the inherent characteristics of the crowdsourcing system that the guided crowdsourcing solution needs to respect. They can include: the number of micro-tasks that each worker is allowed to undertake in a given amount of time, the ability of the system to bargain or not with the workers for the price that each micro-task pays, the ability to interrupt a worker with a new task in case this suits better the objectives of the system, as well as many others.

Coordination Algorithm. As also mentioned in the definition of guided crowdsourcing, the coordination algorithm is what distinguishes the guided from a simple, standard crowdsourcing system. Overall, and following the problem formulation set above, the target of the coordination algorithm is to fulfill the objectives of the crowdsourcing system, for the amount of jobs requested, with the available workers, while respecting the crowdsourcing systems constraints. The coordination algorithm therefore is in fact an optimization technique, over the global performance of the crowdsourcing system. Depending on the exact parameterization made on each of the previous steps, different methods can be used for the design of the coordination algorithm, including queue theory, mechanism design or resource allocation, as described in the related literature section that follows.

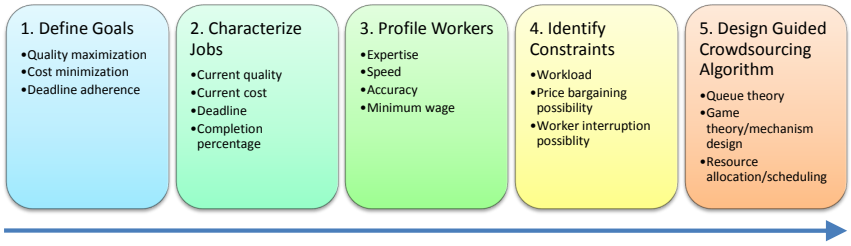

Fig. 1. The process of engineering a guided crowdsourcing solution comprises 5 main steps.

3 Related literature: current trends in guided crowdsourcing algorithms

A queue theory-based analytical method is proposed in [3], for optimizing crowdsourcing in terms of cost. This work focuses on open crowdsourcing, with very high worker and job arrival rates, in which cost is measured in terms of task loss, i.e. those that upon arrival find no available workers to undertake them. The objective of the algorithm in this case is to calculate the optimal cost tradeoff between artificial worker retainment (paying workers to remain in the system until a task arrives) and task loss.

Game theoretic approaches are also used for the design of mechanisms that will optimize the functionality of markets with strategic resources, like the ones that emerge in crowdsourcing applications, in terms of cost. Indicative of this research stream, the works of Ghosh et al. [5] and Archak et al. [1] examine the conditions under which the implementation of contest-based mechanisms among users can reduce cost in crowdsourcing environments

Finally, resource scheduling and allocation approaches have also started to be used for the collective performance improvement of crowdsourcing systems. Psaier et al. [13] examine the improvement of task assignment in crowdsourcing environments by combining a hard/soft resource scheduling algorithm with a mediator responsible of monitoring user skills, organizing activities, settling agreements and scheduling tasks. The algorithm assumes a push crowdsourcing model, i.e. it actively sends requests to workers for the crowdsourcing tasks that need to be completed. Results obtained through simulation for various scenarios show that the algorithm produces better quality in comparison to plain random scheduling, while keeping overall task load within the set limits. In a similar spirit, Khazankin et al. [8] work with scheduled crowdsourcing, in QoS (Quality of Service)-sensitive processes. In their approach an algorithm receives tasks from ordering customers, negotiates with them for quality and temporal job requirements and once an agreement is reached, it distributes the job tasks to appropriate members from the crowd pool. Results of examining the prediction algorithm in a simulated crowdsourcing environment showed that it can efficiently predict the quality capabilities of the crowdsourced workers and therefore provide ordering customers with satisfactory quality guarantees

4 An application study for corporate environments

4.1 Corporate crowdsourcing: A special case with high value potential

Corporate crowdsourcing occurs when crowdsourcing is applied, instead of web workers, to the human network of a company. The main advantage of intra-corporate crowdsourcing is that it permeates the traditional departmental corporate structure, which often hinders the efficient use of human resources. In addition, it is dynamic and it can be used for on-demand tasks that the company might not want to invest with full-time dedicated resources, because of their short term nature. The notion of corporate crowdsourcing can also be extended from intra- to inter-corporate environments, i.e. the borrowing of specialized employees among companies for limited period of time.

The main differences between corporate and open crowdsourcing are three. First, corporate crowdsourcing focuses on knowledge-intensive rather than simple tasks. This is because instead of automatizing simple tasks (such as image recognition), what companies need more from crowdsourcing is to tap the innovation and knowledge creation potential of their human resource employee network [9] (example case: idea gathering for new product development). Indeed crowds can provide a much larger diversity of ideas, compared to individual experts usually hired by companies for knowledge creation, because individuals make their suggestions independently and based on more diverse knowledge backgrounds [12]. Secondly, corporate crowdsourcing allows for lower cost, because the company needs not compete globally with others for the same worker, as it would be the case of hiring external freelancers through open crowdsourcing. Finally, the case of corporate crowdsourcing allows defining a simpler problem setting compared to open crowdsourcing, since here we can assume the presence of a fixed, easier-to-profile pool of workers and ensured worker acceptance on system recommendations. In other words, when contributing people belong to an organization, their profile (including competencies and certain motivation factors) and schedule can be known. Also, constraints such as mandatory time dedicated to contributions to a crowd-sourced problem solving can be imposed by the organization.

All of the above make corporate crowdsourcing platforms more suitable for the guided crowdsourcing approach, since a lot of information can be exploited to actually drive it. In open environments, not only is the information about workers less, but also, they are always free to refuse task assignments. Open crowdsourcing platforms can obviously be extended to gather information from people (e.g. profile, competencies, schedule, etc.), which allows guiding. However, the motivation aspect, as a means to ensure participation, needs to be taken much more into consideration in the open than in the corporate crowdsourcing case.

4.2 Problem instantiation and modeling

The following environment instantiates the generic guided crowdsourcing process to the specificities of a corporate crowdsourcing problem. We model the following elements:

Jobs. $J = \{j_1, \ldots, j_{|J|}\}$ The jobs to be crowdsourced. Each job j_i comprises a:

- Set of n micro-tasks. Each micro-task has a quality q_j, measured in the [0,1] scale, with 1 meaning perfect quality and 0 meaning no quality at all.

- Quality Q_j, the quality of the job calculated as the average quality of its micro-tasks: $Q_i = E[q_j]$, $j \in [1, |J|]$.
- Maximum cost limit C_j, which the enterprise is willing to pay for the jobs accomplishment. The cost of each job is initialized randomly at the beginning of the simulation.

Finally, each job belongs to one of $D_j \in D = \{D_1, D_2, \ldots, D_{|D|}\}$ "expertise domains", with each domain indicating a specific category of corporate knowledge.

Workers. We assume a population of $K = \{k_1, k_2, \ldots, k_{|K|}\}$ workers, who model the employees of the corporation participating in the crowdsourcing system. In contrast to open crowdsourcing we do not assume an infinite crowd but a large but finite pool of people. Each user k_i has:

- Expertise e_i, in each of the simulated expertise domains, measured in the [0,1] range. The quality that an employee contributes to a task is equal to his expertise on the task's domain.
- Speed s_i, i.e. time needed to accomplish a task per domain.
- Minimum "wage" w_i, below which the worker does not accept a micro-task. Since we assume a corporate environment, this wage is not necessarily monetary, but can also be "points" translatable into performance bonuses, days off or other "gamification related" rewards like charity from the part of the company.

Goals. The objectives that the guided crowdsourcing system needs to fulfill, for the specific problem setting, are to:

- Maximize the average quality of the accomplished jobs: $O_1 = max_{j \in |J|} E[Q_j]$
- Minimize the average paid cost: $O_2 = min_{j \in |J|} E[C_j]$

Constraints/Organizational Policies. The constraints depend on the organizational policies that the involved corporation(s) need to pose. For the modeled problem setting we define the constraints of:

- Maximum price. The total paid for each job cannot surpass the maximum cost set for the job: $\Sigma_{i=1}^{m} w_i \leq C_j$, $\forall j \in |J|$, $m \leq n$, where m is the total number of workers that have been given one of the n micro-tasks of the job,
- Non-preemptiveness, i.e. once an employee is occupied with a task they do not enter the system, i.e. they cannot be interrupted to undertake a new task.

Implemented guided crowdsourcing algorithm We propose a guided crowdsourcing algorithm, which uses resource scheduling to dynamically assign micro-tasks to employees according to their individual expertise and inter/intra wage. Given the objectives and constraints of the specific scenario setting (cost minimization and quality maximization), the algorithm suggests to each worker the tasks that pay less from the expertise domain that the worker is mostly expert at. Partially complete jobs (those with at least one task completed) are also preferred, starting from those with the least completion percentage, to boost job completion rate. The above problem modeling is summarized in Table 1.

Problem element	Value
Goal	Maximize average job quality Minimize total cost
Workers	Expertise Speed Minimum accepted wage
Jobs	Number of micro-tasks Quality Cost
Constraints	Maximum job cost limit Non-preemptiveness

Guided crowdsourcing algorithm	For every worker that arrives:
	{
	1. Rank domains of users expertise in descending order
	2. Select first domain on list
	3. Select partially completed jobs from that domain
	4. Rank the tasks of the selected jobs in ascending cost order
	5. Allocate first task on the list
	}

Table 1. The corporate scenario problem, modeled as an instantiation of the generic 5-step guided crowdsourcing process.

4.3 Evaluation

First we parameterize the variables of the above problem modeling (Table 2). Corporate crowdsourcing is expected to work mostly with knowledge-intensive rather than simple tasks, as mentioned above. Therefore, for the selection of the number of users, worker and job arrival rates and total simulation time, we data-mine a real-world system focusing on the crowdsourcing of knowledge-intensive tasks, namely the Data Hub[2]. The extracted dataset covers a timespan of 67 months and features the contributions of 1600 users, and therefore so we set the simulation time equal to 67 simulation units and the simulated population to 1600 workers. Worker expertise is initialized using a beta distribution function, calibrated so that, for each domain, few people are experts and there is long tail of semi or non-experts, as it is the typical case of expertise distribution in enterprise corpora [2]. Worker speed per domain is initialized randomly through a uniform distribution. Worker wage is linearly analogous to expertise (i.e. the more expert a worker, the higher wage they require to fulfill a task). Wage also depends on whether the employee will work for his company (intra-crowdsourcing) or as external worker for another company (inter-crowdsourcing). For intra-crowdsourced work it is set equal to ones expertise, while it doubles if the worker is externally hired.

The interaction of workers with jobs is performed as follows: Workers enter the crowdsourcing platform with an arrival rate λ and jobs are generated with a generation rate μ. Both rates increase exponentially with time. As soon as a worker enters the platform they select a micro-task to fulfil. This selection depends on whether the simulated

[2] http://datahub.io/

system works under a guided or an unguided crowdsourcing manner. In the unguided version, which serves as our benchmark, people seek to maximize their individual profit and therefore they select the task that pays the most, from the ones that surpass their minimum requested wage. In the guided version of the system they can select only among tasks that are recommended to them by the guided crowdsourcing algorithm.

Parameter	Value
Simulation time	67 simulation units
Workers	1600
Domains	20
Tasks per job	3
Job arrival rate μ	$\alpha \cdot e^{\beta t}$, with $\alpha = 130$ and $\beta = 0.05$
Maximum job cost	[0, 2] according to uniform distribution
Worker arrival rate λ	$\gamma \cdot e^{\delta t}$, with $\gamma = 30$ and $\delta = 0.05$
Worker wage	ρ·expertise, with $\rho = 1$
Worker speed	[0, 1], according to uniform distribution
Worker expertise	[0,1] according to beta distribution
Worker wage	α·expertise, with $\begin{cases} \alpha = 1, \text{ if internally hired} \\ \alpha = 2, \text{ if externally hired} \end{cases}$
Companies	50 (for the inter-corporate scenario)

Table 2. Parameters used for the evaluation.

Scenario 1. Intra-corporate crowdsourcing In the first scenario all employees belong to the same organization. We examine the performance of the guided crowdsourcing algorithm according to four criteria: average quality, cost, completed versus started tasks, and time until completion (Fig. 2a, with all results normalized to the [0,1] scale). As it can be observed, it performs better than the unguided system in terms of quality (0.86 instead of 0.23) and cost (0.72 instead of 0.93), which are the two criteria that the algorithm is designed to optimize. The guided crowdsourcing algorithm also achieves to keep the completion rate of finished versus started jobs at comparable levels (0.92) with that of the unguided system (0.98). However, the above come at the cost of timeliness (0.56 instead of 0.18 average time units), since the algorithm gives each worker the task that he is most expert at, therefore "spreading" user contributions across jobs, in comparison to the unguided system where users all target the same, high-paying jobs.

Scenario 2. Inter-corporate crowdsourcing In the second scenario, workers belong to multiple companies, which can lend them to one another. In this case, for each worker we assume two wages: an intra-corporate one, equal to the workers expertise, and an inter-corporate one, which is double the intra-corporate one. Accordingly, each company has an upper limit for the percentage of employees it can borrow externally. We simulate 50 companies and examine the average quality gained and the extra cost paid, for different upper employee borrowing limits. As it may be observed, and intuitively expected, the more employees a company borrows the more qualitative tasks it achieves, but at a higher cost (Fig. 2b). Therefore, although guided crowdsourcing can be used to augment job quality, often significantly, it remains at the disposal of each organization, to determine the best tradeoff suitable for its crowdsourcing needs, according to its needs, expertise availability and cost constraints.

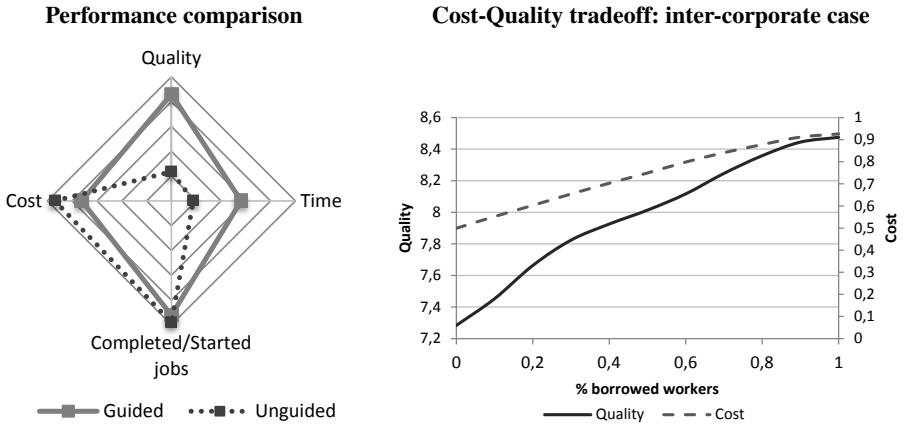

Fig. 2. a) Performance comparison between the examined guided crowdsourcing algorithm vs. unguided crowdsourcing. All axes are presented in % ratios in respect to their maximum value. b) The tradeoff between cost and quality for inter-corporate crowdsourcing.

5 Conclusion and Perspectives

Guided crowdsourcing is a new, emerging domain with high potential. It refers to the optimization of the performance of crowdsourcing systems, in terms of quality, cost and timeliness, by using AI-based methods to coordinate the involved user crowd. In this paper we present the notion of guided crowdsourcing, formulate the process of engineering a guided crowdsourcing solution as a 5-step process, present the main research streams in the area and examine its potential on the application case of inter and intra corporate crowdsourcing. Results are promising, indicating that guided crowdsourcing can help achieve better performance in comparison to typical unguided crowdsourcing.

A number of topics need to be further investigated. Firstly, in the problem instantiation treated in this paper a fixed pool of workers and jobs is assumed. This may hold true for certain cases, like the corporate one (where the worker/job pool is either fixed or predictable with high accuracy), but in other environments, like open crowdsourcing, there is a need to model the uncertainty in the size and availability of the worker pool, as well as on the load of job demands. In this case, a dynamic scheduling problem formulation and algorithm may be more appropriate. Also, in contrast to corporate environments (where workere expertise can be considered known or easy to obtain, e.g. using data from the employees' job description or previously undertaken tasks within the organization), an extension of the proposed approach to open crowdsourcing environments would necessitate the incorporation of an adaptive skill learning mechanism, such as the one proposed in [10]. Other optimization goals may also be considered, regarding corporate crowdsourcing. For example, instead of maximizing average quality while remaining under a certain cost limit, a company might prefer to minimize the total project cost and keep a minimum quality baseline or optimize a weighted combination of both, which would pose the need to define and solve the problem as multi-objective resource scheduling. The proposed approach can also be compared to more benchmarks, ex-

tending the comparison with the fully unguided benchmark used in this paper. These benchmarks may include the partial filtering of workers based on the quality of their previous contributions (e.g. for certain types of jobs in Amazon Mechanical Turk, the requesters can allow the participation of only certain "qualified" workers). It would be also interesting to compare the proposed approach with the algorithms proposed by other studies of the related literature, and in particular those that use resource allocation as their main algorithmic technique. Finally, further research needs to consider the broader context of integrating guided crowdsourcing in the enterprise, investigating issues related to internal regulation changes that are necessary to accomodate in-house crowdsourcing, ethical issues, as well as the topic of incentinve engineering.

Summarizing, guided crowdsourcing is a technology and research area with significant potential, but also with much room for improvement, both in terms of algorithmic efficacy, as well as in terms of harmonization with the human factor that it entails.

References

1. Nikolay Archak. Optimal design of crowdsourcing contests. *ICIS 2009 Proceedings*, 200(512):0–16, 2009.
2. Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 43–50, New York, NY, USA, 2006. ACM.
3. Michael S. Bernstein, David R. Karger, Robert C. Miller, and Joel Brandt. Analytic methods for optimizing realtime crowdsourcing. *CoRR*, abs/1204.2995, 2012.
4. Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96, April 2011.
5. Arpita Ghosh and Patrick Hummel. Implementing optimal outcomes in social computing: a game-theoretic approach. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 539–548, New York, NY, USA, 2012. ACM.
6. Panagiotis G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS*, 17(2):16–21, December 2010.
7. Panagiotis G Ipeirotis and John J Horton. The need for standardization in crowdsourcing. *CHI 2011 Crowdsource workshop*, pages 1–4, 2011.
8. Roman Khazankin, Daniel Schall, and Schahram Dustdar. Predicting qos in scheduled crowdsourcing. In *Proceedings of the 24th international conference on Advanced Information Systems Engineering*, CAiSE'12, pages 460–472, Berlin, Heidelberg, 2012. Springer-Verlag.
9. Aniket Kittur. Crowdsourcing, collaboration and creativity. *XRDS*, 17(2):22–26, December 2010.
10. Xuan Liu, Meiyu Lu, Beng Chin Ooi, Yanyan Shen, Sai Wu, and Meihui Zhang. Cdas: a crowdsourcing data analytics system. *Proc. VLDB Endow.*, 5(10):1040–1051, June 2012.
11. Ioanna Lykourentzou, Katerina Papadaki, Dimitrios J. Vergados, Despina Polemi, and Vassili Loumos. Corpwiki: A self-regulating wiki to promote corporate collective intelligence through expert peer matching. *Inf. Sci.*, 180(1):18–38, January 2010.
12. Marion K. Poetz and Martin Schreier. The value of crowdsourcing: Can users really compete with professionals in generating new product ideas? *Journal of Product Innovation Management*, 29(2):245–256, 2012.
13. H. Psaier, F. Skopik, D. Schall, and S. Dustdar. Resource and agreement management in dynamic crowdcomputing environments. In *Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International*, pages 193–202, 2011.