# Optimal User Choice Engineering in Mobile Crowdsensing with Bounded Rational Users

Merkouris Karaliopoulos and Iordanis Koutsopoulos
*Department of Informatics*
*Athens University of Economics and Business*
Athens, Greece
email: {mkaralio,jordan}@aueb.gr

Leonidas Spiliopoulos
*Center for Adaptive Rationality*
*Max Planck Institute for Human Development*
Berlin, Germany
email: spiliopoulos@mpib-berlin.mpg.de

***Abstract*—In mobile crowdsensing (MCS), users are repeatedly asked to make *choices* between a set of alternatives, *i.e.*, whether to contribute to a task or not and which task to contribute to. The platform coordinating the MCS campaigns *engineers* these choices by selecting the tasks to present to each user and offering incentives to ensure user contributions and maximize the benefit from them. In this paper, we revisit the well-investigated question of how to optimize the contributions of crowds of mobile end users to MCS tasks. However, we depart from the bulk of related literature by explicitly accounting for the *bounded rationality* of human decision making. Bounded rationality is a consequence of cognitive and other kinds of constraints, (*e.g.*, time pressure) and has been studied extensively in behavioral science.**

**We model bounded rationality after two instances of lexicographic decision-making models that originate in the field of cognitive psychology: Fast-and-Frugal-Trees (FFTs) and Deterministic Elimination by Aspects (DEBA). With each MCS task modeled as a vector of feature values, the decision process under both models proceeds through sequentially parsing lexicographically ordered features, resulting in choices that are satisfying, but not necessarily optimal. We study, in particular, scenarios where a single task or a pair of tasks are presented to MCS users together with reward offers that adhere to per-task budget constraints. We formulate the optimization problems that emerge for the MCS campaign organizers as instances of the Generalized Assignment Problem (GAP), an NP-hard problem for which approximate algorithms are available. Our evaluation suggests that our optimization approach exhibits significant gains when compared to heuristic rules that do not account for the lexicographic structure in human decision making.**

***Index Terms*—Mobile crowdsensing; incentive allocation; bounded rationality; task recommendation; decision trees; user choice engineering.**

## I. Introduction

Mobile crowdsensing (MCS) has generated many expectations over the last decade that it will transform the way information is generated and shared among parties interested in it [1]. Technically, it couples the enhanced sensing capabilities of smart mobile devices with a variety of mobile and social technologies, which have made the uploading, processing, and sharing of data easier than ever before. As a result, MCS has been applied in diverse application areas such as environmental monitoring[1], transportation[2], and participatory governance[3].

[1]NoiseTube, http://www.noisetube.net/index.html.
[2]Waze. Outsmarting Traffic Together, http://www.waze.com/.
[3]SeeClickFix, https://seeclickfix.com/pages/311-app.html.

The implementation of MCS campaigns is more often than not coordinated by a mobile platform, end users typically interacting with it through a mobile frontend (app). It is a mission of this platform to identify those users who are most suitable for the MCS tasks at hand. These tasks may be diverse in scope such as *e.g.*, submitting measurements to create transportation or pollution maps, traveling across the city to capture photos from points of interest, submitting expert opinions, sharing data or advice for lifestyle and healthcare improvement, delivering parcels, and so on. Users carrying out a task bear some cost in terms of time, cognitive resources, device battery, or physical distance that needs to be covered to perform the task. Hence, they need to be provided with incentives in order to get engaged in an MCS campaign and make contributions that serve its objective(s). Incentives are typically monetary or in-kind rewards and their efficient allocation to potential contributors demands a good understanding of their particular interests and preferences and how they decide whether to contribute or not to tasks. Such *user profiling* information can be collected in different ways: implicitly, from historical data about past users' interaction with the MCS platform or their activity in other social media sites [2]; or, explicitly, through customized questionnaires built in the app.

User profiling comes hand-in-hand with a modeling hypothesis about how users reach decisions, *e.g.*, whether to contribute or not to a given task, and how they make choices, *e.g.*, which task to contribute to out of a set of recommended alternatives. In this respect, the de facto assumption in the current MCS literature is that MCS users behave as *fully rational* agents [3]. Namely, they exhaustively enumerate all information available at hand to (strategically) optimize an implicit utility function that quantifies the "net value" of their contributions to different tasks.

Our point of departure in this work is that users are *bounded rational* agents. This term was used by Herbert A. Simon in [4] to denote that decision makers can rarely know and evaluate all possible outcomes of their decisions with sufficient precision due to constrained memory and processing capacities, and limited or imperfect information about the decision environment. Since then, extensive experimental evidence has been accumulated *e.g.*, in [5] [6], suggesting that human decisions seek to *satisfice*[4] rather than optimize. In parallel, modeling work in cognitive psychology, behavioral economics and marketing science, has come up with modeling constructs that aim to match this evidence [7] [8].

Our paper leverages this volume of work to mark a radical departure from standard modeling practice in MCS literature. To the best of our knowledge, it is the first attempt to import knowledge and modeling tools from behavioral science to the problem of task recommendation and incentive allocation in MCS. In our modeling approach, presented in section II, there is a finite feature space and MCS tasks are represented as feature vectors in this space. User are heterogeneous with respect to preferences and interests, hence they typically rank these features differently in order of importance.

Then, in section III, we model the bounded rationality aspect after two instances of lexicographic decision-making models that originate in the field of cognitive psychology: Fast-and-Frugal-Trees (FFTs) and Deterministic Elimination by Aspects (DEBA). Under both models, the task selection process proceeds through sequentially parsing the lexicographically ordered features. We consider, in particular, scenarios, where the MCS platform provides users with a single task or a pair of tasks and prompts their contributions offering rewards that are subject to per-task budget constraints.

Our main contribution comes in section IV. Therein, we address the respective *joint task recommendation and incentive reward allocation problem* faced by an MCS platform that aims to maximize the aggregate quality of user contributions to MCS tasks. We formulate these optimization problems as instances of the Generalized Assignment Problem (*e.g.*, see chapter 7 in [9]), an NP-hard problem for which approximate algorithms are available.

Then, in section V, we compare the solutions that account for the bounded rationality in the choices of users against four heuristics that ignore it. Our results demonstrate significant performance gains over a broad range of system settings and user profile mixes. In section VI, we contrast our work against the existing literature. We conclude the paper in section VII highlighting directions for future work.

## II. System model

Our model involves task-issuing entities which issue crowdsensing tasks, users who may choose to contribute to these tasks, and a MCS platform with a mobile frontend (UI) that facilitates the coordination of the two parties, the task issuers and the potential task contributors.

Formally, let $\mathcal{U}$ be the set of potential data contributors, with $U = |\mathcal{U}|$ denoting the set cardinality. This set consists of mobile users with smart devices who have registered with the platform and run the respective app on their devices. Let also $\mathcal{M}$ be the set of all tasks that are managed by the platform and $\mathcal{M}_u, u \in \mathcal{U}$ be the set of crowdsensing tasks that are eligible for allocation to user $u$. In general, this set of tasks varies with time, depending on the user's location, and possibly other contextual information collected by the app. Herein, we assume that the task allocation process is carried out at distinct epochs over a given snapshot of the system (advertised tasks, user locations); hence, the set of tasks $\mathcal{M}_u$ is considered fixed for each user.

### A. Tasks as feature vectors

Each task $m \in \mathcal{M}$ may be represented as a vector of values on a set of $L$ *features*, $\mathbf{f}_m = (f_m^1, \dots, f_m^L)$. These features may be numerical or categorical and reflect different aspects of the task such as the reward offered for contributing to it that may be monetary or in-kind (*e.g.*, a coupon); the physical location, where the task needs to be carried out; the average time or effort it takes to perform the task; the battery/computational requirements posed by the task; and the context, *e.g.*, commercial or nonprofit, of the service that is facilitated by the requested task contribution. For simplicity of exposition, we assume in the sequel that user choices are determined by $L = 2$ features, the task reward serving as *incentive* for contributing to the task, and the task distance representing the *cost* a user incurs when contributing to a task.

Typically, each task $m$ comes up with some budget $B_m$. This sets an upper bound on what the task-issuing entity is willing to spend on rewarding task contributors.

### B. User choices as a multi-attribute decision-making problem

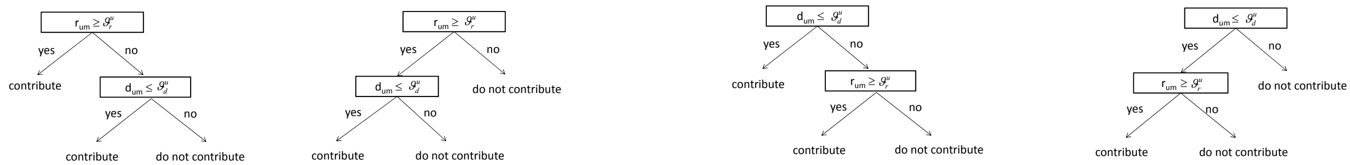
The crowdsensing campaign designer interacts with users through the mobile app and notifies them about specific tasks as they move around. An example of such a task notification could be: *"Coffee place $m$, at distance $d_m$ from your current location, offers a reward $r_m$ for taking a couple of nice photos of it. Click on the offer to accept it, upload the photos, and get your voucher."*

All users are not equally qualified for a task. We denote with $q_{um}$ the quality of contribution a user $u$ can make to task $m$. Practically, this may relate to the interests and skills of $u$. In the aforementioned example, an amateur or professional photographer would take better photos than someone who does not practice photography at all. It could also relate to the equipment the user possesses, *e.g.*, a smartphone with a higher resolution camera.

Each task feature (interchangeably called *cue* hereafter) ranks differently the tasks in $\mathcal{M}_u$ for a given user $u$, depending on her own preferences. The way individual users prioritize/weigh the different cues can be inferred from historical data, that is, data about past choices they have made when tasks were offered to them. Yet, these same data can be combined with different modeling hypotheses about the way users decide. *The crucial modeling hypothesis in this work is that users exhibit bounded rationality*. In what follows, we develop this hypothesis proposing a specific family of heuristic decision-making models that capture the bounded

[4]Satisfice is a "portmanteau" word of satisfy and suffice and implies searching through available alternatives until one is deemed acceptable.

[4]Terms "feature", "cue", "aspect" are used interchangeably in different knowledge areas such as data science, cognitive psychology and marketing.




a. FFT1 (lenient pectinate, $u \in \mathcal{U}_r$) b. FFT2 (strict pectinate, $u \in \mathcal{U}_r$) c. FFT3 (lenient pectinate, $u \in \mathcal{U}_d$) d. FFT4 (strict pectinate, $u \in \mathcal{U}_d$)

Fig. 1. The four possible FFTs that emerge for different users, depending on whether they prioritize the reward ($\mathcal{U}_r$) or the distance ($\mathcal{U}_d$) cue in their choices.

rationality aspect, such as the Fast-and-Frugal Trees (FFTs) and the Deterministic Elimination by Aspects (DEBA) model.

## III. Users as bounded rational agents

### A. *Fast-and-frugal heuristics: background*

Cognitive heuristics are *descriptive*, as opposed to *normative*, models of human decision-making. Namely, they provide more direct implications for the cognitive processes that guide a decision/choice, focusing more on how humans *do* decide rather than how they *should* decide [4] [6]. An important subclass of these heuristics are the *lexicographic* heuristics [5]. Common to them is that the features describing different alternatives are inspected in a *user-specific* order, and one of the alternatives is chosen upon the first feature that discriminates between task alternatives. Hence, the user does not exhaustively process all information at hand (*i.e.*, the full feature set) to make a choice.

Three major stylized facts have emerged with respect to the accuracy of lexicographic, and more generally cognitive heuristics, in predicting human decisions [10]. First, their accuracy is comparable to that of more complex and computationally-demanding models such as regression, neural networks, classification and regression trees (CARTs), or naive Bayes. Second, these simple heuristics often perform better in predicting out-of-sample human choices when compared to matching past choices. Third, each model outperforms the other under certain conditions, since each one is capable of exploiting different properties of the decision environment such as the correlation between different features. The coupling between the structure of choice rules and the context (choice environment) allows the former to efficiently exploit the latter, in what is called *ecological rationality*.

*1) Fast-and-frugal trees:* Fast-and-Frugal Trees (FFTs) are one instance of cognitive lexicographic heuristics [11] [12]. They are deterministic *binary* decision trees. As with all lexicographic heuristics, the cues of the alternatives are first ordered. Then, at each level of the tree, one cue is inspected. The context of the inspection depends on the type of the cue, binary or continuous. For binary cues, "1" typically denotes existence of a cue and "0" its absence. Continuous cues are compared against satisfiability thresholds: a cue may favor a decision alternative if it exceeds the threshold or if it does not meet it, depending on whether a cue is positively (*e.g.*, MCS task rewards) or negatively (*e.g.*, MCS task distance) correlated with an alternative (*e.g.*, contribute to the task). Both the order of the cues and the satisfiability thresholds are learnt while training the model with a sample of past user decisions (training sample). In [12], these thresholds are set to the medians of the cue values in the training sample, whereas in [13] they are computed with enumeration over the whole range of cue values.

A key feature of FFTs is that at every level of the tree, at least one of the inspection outcomes results in a decision, *i.e.*, at least one of the child nodes is a leaf node. They are called *frugal* because they end up processing limited information; and they are *fast*, because this non-exhaustive information processing accelerates the decision-making process.

FFTs were originally proposed for *inference* tasks. Thereby, one alternative is the correct one according to an objective criterion and the goal is to figure out which one is this by inspecting different cues. A well known experiment in this respect is the one concerning the size of German cities [7]. Users are presented with pairs of German cities and are asked to infer which one is larger in terms of population given how they score in a number of cues such as whether they host a major exhibition or are home to a soccer team in the German major soccer league.

In our paper, we use FFTs in a *preferential choice* context. In such tasks, there is no objective "ground truth" as in inferential tasks above. The ordering of cues and the respective satisficing thresholds depend on the subjective preferences and priorities of *each* user and are learnt out of separate training processes.

*2) Deterministic Elimination by Aspects (DEBA):* The DEBA heuristic in [19] also ranks cues in order of decreasing importance ($x_1$, $x_2$, ...). When the cues are discrete, the alternatives are readily codified as ordered sequences of ones and zeros. DEBA then inspects the value of all alternatives on $x_1$ and eliminates all alternatives featuring a zero on it. The process is repeated when parsing the second and the remaining attributes until a single alternative remains. If more than one alternatives are left after all cues are inspected, or if there is a cue eliminating all remaining alternatives, a choice is made randomly among the currently surviving ones.

DEBA can also work with alternatives featuring continuous cues as long as these are digitized according to some rule before being processed by the heuristic. The simplest and most used such rule is called median split [5]. For each cue $x_i$, the rule computes the median $med_i$ of the cue values across all alternatives. In alternatives, where the cue value exceeds $med_i$, the cue is codified as 1(0) for favorable(adverse) cues and vice versa. Hence, the median value $med_i$ implicitly quantifies the satisficing threshold for cue $x_i$.

In the remainder of the section, we explain how these two lexicographic heuristics (FFTs and DEBA) model the user choices in app scenarios that recommend one task at a time to MCS users (*single-task offers*) and others that present them with pairs of task offers (*paired task offers*).

### *B. Single-task offers: user decisions as fast-and-frugal trees*

Consider the scenario where a single task is recommended to each user through the app and the user may accept the offer or reject it. For the case of $L = 2$ cues, the decision of user $u$ to make or not a task contribution is determined by two user-dependent *acceptability thresholds* $\theta_r^u$ and $\theta_d^u$ pertaining to the task reward and physical distance, respectively. In particular, the user decision process is modeled by one out of a set of the four different FFTs shown in Fig. 1.

The four trees differentiate regarding the order of cue inspection and whether the exit upon a cue inspection is triggered when an acceptability threshold is satisfied or not. The two leftmost trees, FFT types 1 and 2, model users who prioritize the task reward in their decisions. FFT1 models users that decide to contribute to a task if its reward $r_{um}$ exceeds the reward acceptability threshold, *i.e.*, $r_{um} \geq \theta_r^u$, while FFT2 models users who decide to contribute to the task if it satisfies both the reward and distance acceptability thresholds *i.e.*, if $r_{um} \geq \theta_r^u$ and $d_{um} \leq \theta_d^u$. On the other hand, the two rightmost FFTs, FFT types 3 and 4, represent users who prioritize the distance of offered tasks. In FFT3 it suffices that $d_{um} \leq \theta_d^u$ in order for the user to contribute to the task, while in FFT4 both $d_{um} \leq \theta_d^u$ and $r_{um} \geq \theta_r^u$ need to hold. Note that FFT1 and FFT3 feature an exit at the *first* tree level in favor of contributing to a task, when the respective thresholds, $\theta_r^u$ or $\theta_d^u$ are satisfied.

All four FFTs are of the *pectinate* (or *rake*) type [12]; namely, one of the two alternatives is chosen under a strong conjunction rule. FFT2 and FFT4 model users who decide to contribute to a task only when both acceptability thresholds are satisfied (*strict pectinates*). On the other hand, under FFT1 and FFT3 the user decides *not* to contribute to a task only when both thresholds are violated (*lenient pectinates*); equivalently, the satisfaction of one of the two thresholds suffices for a positive response to the task offer.

We denote by $\mathcal{U}_i$, $i = 1, 2, 3, 4$ the subset of users modeled by FFT$i$, and let $\mathcal{U}_r = \mathcal{U}_1 \cup \mathcal{U}_2$ and $\mathcal{U}_d = \mathcal{U}_3 \cup \mathcal{U}_4$. Thus, $\mathcal{U}_d, \mathcal{U}_r$ denote the subsets of users who place priority on the distance and the reward feature respectively, and form a partition with $\mathcal{U}_d \cap \mathcal{U}_r = \emptyset$ and $\mathcal{U}_d \cup \mathcal{U}_r = \mathcal{U}$. These subsets and the user-dependent thresholds are the outcome of the model training process (*e.g.*, [13]).

### *C. Paired task offers: user choices as DEBA processes*

Now, assume that the mobile app offers two alternatives (MCS tasks) $m_1$, $m_2$ to each user $u$, specified by the reward-distance pairs $(r_{um_1}, d_{um_1})$ and $(r_{um_2}, d_{um_2})$. The user may choose to contribute to one of the two tasks recommended by the app or decline both offers.

TABLE I

CODING A TASK OFFER $(r_{um}, d_{um})$ TO USER $u$ AS A DEBA ALTERNATIVE $(m_{u1}, m_{u2})$ WITH ACCEPTABILITY THRESHOLDS $\theta_r^u$, $\theta_d^u$.

| | $u \in \mathcal{U}_r$ | $u \in \mathcal{U}_d$ |
|---|---|---|
| $m_{u1}$ | 1 if $r_{um} \geq \theta_r^u$<br>0 if $r_{um} \leq \theta_r^u$ | 1 if $d_{um} \leq \theta_d^u$<br>0 if $d_{um} \geq \theta_d^u$ |
| $m_{u2}$ | 1 if $d_{um} \leq \theta_d^u$<br>0 if $d_{um} \geq \theta_d^u$ | 1 if $r_{um} \geq \theta_r^u$<br>0 if $r_{um} \leq \theta_r^u$ |

TABLE II

VALUE RANGES OF THE REWARD AND DISTANCE CUES, FOR WHICH DIFFERENT FFT MODELS ACCEPT OFFERS AND CONTRIBUTE TO TASKS.

| | $r_{um} \geq \theta_r^u$ | $r_{um} < \theta_r^u$ |
|---|---|---|
| $d_{um} < \theta_d^u$ | FFT1-FFT4 | FFT1, FFT3 |
| $d_{um} \geq \theta_d^u$ | FFT1, FFT3 | – |

When users invoke the DEBA model in making choices, they treat each alternative as a string $m = (m_{u1} m_{u2})$ of ones and zeros according to Table I. It is also convenient to consider the option of contributing to neither of the two tasks as a third virtual task alternative, which is coded by $m_{0,r} = (01)$ for users $u \in \mathcal{U}_r$, prioritizing the task reward in their choices, and by $m_{0,d} = (10)$ for users $u \in \mathcal{U}_d$, first considering the task physical distances. The model training process infers (*i*) whether $u \in \mathcal{U}_r$ or $u \in \mathcal{U}_d$; (*ii*) the threshold values, $\theta_r^u, \theta_d^u$.

## IV. PROBLEM FORMULATION

As mentioned earlier, for illustrative purposes we consider that individual task choices are described by two features: the task distance, which has cost/effort dimensions for a user and is tied to a given task; and, the task reward, which reflects the user benefit out of a contribution and can be controlled by the MCS platform.

The MCS campaign designer, then, aims at engineering the MCS task offers that are made to users. This implies choosing the tasks and the accompanying rewards offered to each MCS user in ways that induce task contributions, which are beneficial for the overall campaign objectives. For instance, to users who prioritize based on exerted effort (*e.g.*, distance), the app may choose to recommend tasks that lie at short distance, offering small rewards. It could then direct these savings on the task budget to expert users who prioritize based on the offered task rewards. Thus, the budgets of the various tasks could be managed efficiently to attract more contributions by better qualified users. In what follows, we describe how this objective can be formalized under single- and paired-task offers. In both cases, we assume that the MCS platform objective is to maximize the aggregate quality of attracted task contributions in light of the per task budget constraints posed by the task-issuing entities.

### *A. Single-task offers*

A key remark in this case is that the four different FFTs in Fig. 1, which model how users decide, can be grouped in two categories according to how users respond to a given

$\{reward, distance\}$ task offer. As shown in Table II, users modeled by the strict pectinates (either FFT2 or FFT4) will accept an offer as long as both the reward and distance threshold conditions are satisfied. The hint for the platform provider is that those users will never accept offers for tasks at distances greater than $\theta_d^u$, whereas a reward marginally beyond $\theta_r^u$ will suffice to ensure their contributions to tasks that lie closer than $\theta_d^u$. Hence, the set of eligible tasks for a user $u \in \mathcal{U}_2 \cup \mathcal{U}_4$ is:

$$\mathcal{M}_u = \{m : d_{um} \leq \theta_d^u\},\ u \in \mathcal{U}_2 \cup \mathcal{U}_4 \,. \tag{1}$$

On the other hand, for users $u \in \mathcal{U}_1 \cup \mathcal{U}_3$ modeled by the lenient pectinates, all tasks may be of interest depending on the reward that is offered for them. A reward at least as high as $\theta_r^u$ will have to be offered to them for tasks further than $\theta_d^u$, whereas a much smaller reward would suffice for tasks within distance $\theta_d^u$. Therefore,

$$\mathcal{M}_u = \{m : d_{um} \leq D_{max}\},\ u \in \mathcal{U}_1 \cup \mathcal{U}_3 \,. \tag{2}$$

where, $D_{max}$ is an upper bound on how remote are the tasks offered to users by the MCS platform.

Hence, the MCS platform provider can ensure a contribution from user $u$ to task $m$ by targeting the offered task rewards as follows:

$$r_{um} \geq \begin{cases} \theta_r^u, u \in \mathcal{U}_2 \cup \mathcal{U}_4, d_{um} \leq \theta_d^u \\ \theta_r^u, u \in \mathcal{U}_1 \cup \mathcal{U}_3, d_{um} > \theta_d^u \\ r_{min}, u \in \mathcal{U}_1 \cup \mathcal{U}_3, d_{um} \leq \theta_d^u \end{cases} \tag{3}$$

where $r_{min}$ is a minimum constant reward a priori set by the platform for all task contributions.

Then, the MCS campaign organizer seeks to identify which task $m$ to recommend to each user $u$, with reward $r_{um}$ that ensure their positive responses and contributions, in line with (3), and are subject to the budget constraints for each task. If $\mathbf{x} = (x_{um} : u \in \mathcal{U}, m \in \mathcal{M}_u)$ denotes such a task assignment, with $x_{um} = 1$ if user $u$ is assigned task $m$, and 0 otherwise, the optimization problem (P1) faced by the MCS platform can be written as follows:

$$\max_{\mathbf{x}} \quad \sum_{u \in \mathcal{U}} \sum_{m \in \mathcal{M}_u} q_{um} x_{um} \,, \tag{4}$$

$$s.t. \quad \sum_{u : m \in \mathcal{M}_u} r_{um} x_{um} \leq B_m \quad \forall m \in \mathcal{M} \tag{5}$$

$$\sum_{m \in \mathcal{M}_u} x_{um} = 1 \quad \forall u \in \mathcal{U} \qquad (P1) \tag{6}$$

$$x_{um} \in \{0, 1\} \quad u \in \mathcal{U}, m \in \mathcal{M}_u \,. \tag{7}$$

In (5), reflecting the per task budget constraints, the rewards $\{r_{um}\}$ are taken to be the minimum ones satisfying (3)[5]; it is trivial to show that the MCS platform cannot increase the objective function value (4) by offering rewards beyond the minimum needed. Equation (6) codifies that exactly one MCS task has to be recommended to each user.

The problem (P1) is an instance of the maximum Generalized Assignment Problem (GAP) (see, for example, [9]). Recommended tasks in (P1) correspond to bins in the generic GAP typology, task budgets to bin capacities, user contributions to items, expected qualities of user contributions to item profits, and rewards offered for task contributions to item sizes. The GAP is an NP-hard problem. Approximate algorithms have been proposed for it in [14], [15] and [16], where the best known $\frac{e}{e-1} + \epsilon$-approximation is obtained by a Linear Programming based algorithm.

### *B. Paired task offers*

In this case, we consider apps that recommend pairs of tasks to users, giving them a choice regarding the task they will contribute to. Consider a user $u$ at the time that the app is about to issue task recommendations. Depending on the task distance acceptability threshold $\theta_d^u$ and whether the user prioritizes distance or reward in her choice, a subset of all possible pairs of tasks in $\mathcal{P}_u = \mathcal{M}_u \times \mathcal{M}_u$ constitute meaningful offers on behalf of the app. For instance, it would make no sense to issue a recommendation to $u$ for tasks $(m_1, m_2)$ with $d_{m_1}, d_{m_2} > \theta_d^u$ if $u \in \mathcal{U}_d$ since, irrespective of the chosen rewards, both tasks would be rejected by $u$.

On the other hand, given a meaningful pair of tasks $p = (m_1, m_2) \in \mathcal{P}_u$ at physical distances $(d_{m_1}, d_{m_2})$ from user $u$, the app can choose the rewards for the two tasks in many different ways in order to direct $u$ towards the one or the other task. These options are summarized in Table III.

In practice, the rewards take discrete values so that their combinations (columns 2 and 4 in Table III) that result in a given task choice for each task pair form a finite set. Even better, as shown in columns 4 and 7 of Table III, the minimum reward that has to be given to a user to induce a contribution to a task equals her reward acceptability threshold. Therefore, for each pair of tasks $(m_1, m_2) \in \mathcal{P}_u$, the original reward range of the app can be narrowed down to a much smaller subset of meaningful paired offers. For example, for the first possible pair of tasks ($d_{um_1} \leq \theta_d^u < d_{um_2}$, first row in Table III), the app directs the user to task $m_1$ by choosing reward $\theta_r^u$ for $m_1$ and one arbitrary but fixed reward for $m_2$, whereas it can direct her to task $m_2$, as far as $u \in \mathcal{U}_r$, by offering reward $\theta_r^u$ for $m_2$ and anything smaller for $m_1$ (see Fig. 2).

Hence, for any user $u$ and pair of tasks $p = (m_1, m_2)$ in $\mathcal{P}_u$, *i.e.*, for given $d_{um_1}$ and $d_{um_2}$, and through properly choosing the offered rewards, the platform can induce one out of a larger set $t(p)$ of possible contributions. Each possible contribution comes at the cost (reward) and with the benefit (contribution quality) shown in columns 4 or 7 of Table III for the MCS campaign. This way, the union of possible task pairs in $\mathcal{P}_u$ yields an inflated set of tuples $\mathcal{T}_u$. Each tuple $t$ uniquely determines the pair of tasks recommended to user $u$, the offered rewards for each task, and, most importantly, the induced task choice of the user, $m(t)$. This in turn, determines the quality $q_{um(t)}$ of the contribution and the payment $r_{um(t)}$

[5]To be more precise, the reward should be $\theta_r^u + \epsilon, \epsilon \to 0$, but we work with $\theta_r^u$ for the sake of simplicity.

TABLE III

POSSIBLE TASK REWARD OFFERS AND RESULTING CHOICES OF USER $u$ WITH ACCEPTABILITY THRESHOLDS $(\theta_r^u, \theta_d^u)$ FOR ALL POSSIBLE PAIRS OF TASKS $(m_1, m_2) \in \mathcal{P}_u$. THE CHOICES ACCOUNT FOR THE USER OPTION NOT TO CONTRIBUTE TO ANY TASK, *i.e.*, VIRTUAL TASKS $m_{0,r}$ AND $m_{0,d}$ AS DEFINED IN III-C. COLUMNS 4 AND 7 LIST THE MINIMUM PAYMENTS THAT ENSURE THE TASK CHOICES IN COLUMNS 3 AND 6, RESPECTIVELY.

| Paired task cases | $u \in \mathcal{U}_r$ | | | $u \in \mathcal{U}_d$ | | |
|---|---|---|---|---|---|---|
| | reward ranges | choice | (contrib. quality, min cost) | reward ranges | choice | (contrib. quality, min cost) |
| $d_{um_1} \leq \theta_d^u < d_{um_2}$ | $r_{um_1} \geq \theta_r^u$, any $r_{um_2}$<br>$r_{um_2} \geq \theta_r^u \geq r_{um_1}$ | $m_1$<br>$m_2$ | $(q_{um_1}, \theta_r^u)$<br>$(q_{um_2}, \theta_r^u)$ | $r_{um_1} \geq \theta_r^u$, any $r_{um_2}$ | $m_1$ | $(q_{um_1}, \theta_r^u)$ |
| $d_{um_2} \leq \theta_d^u < d_{um_1}$ | $r_{um_2} \geq \theta_r^u$, any $r_{m_1}$<br>$r_{um_1} \geq \theta_r^u \geq r_{um_1}$ | $m_2$<br>$m_1$ | $(q_{um_2}, \theta_r^u)$<br>$(q_{um_1}, \theta_r^u)$ | $r_{um_2} \geq \theta_r^u$, any $r_{um_1}$ | $m_2$ | $(q_{um_2}, \theta_r^u)$ |
| $d_{um_1}, d_{um_2} \leq \theta_d^u$ | $r_{um_1} \geq \theta_r^u > r_{um_2}$<br>$r_{um_2} \geq \theta_r^u > r_{um_1}$ | $m_1$<br>$m_2$ | $(q_{um_1}, \theta_r^u)$<br>$(q_{um_2}, \theta_r^u)$ | $r_{um_1} \geq \theta_r^u > r_{um_2}$<br>$r_{um_2} \geq \theta_r^u > r_{um_1}$ | $m_1$<br>$m_2$ | $(q_{um_1}, \theta_r^u)$<br>$(q_{um_2}, \theta_r^u)$ |
| $d_{um_1}, d_{um_2} > \theta_d^u$ | $r_{um_1} \geq \theta_r^u > r_{um_2}$<br>$r_{um_2} \geq \theta_r^u > r_{um_1}$ | $m_1$<br>$m_2$ | $(q_{um_1}, \theta_r^u)$<br>$(q_{um_2}, \theta_r^u)$ | - | - | - |

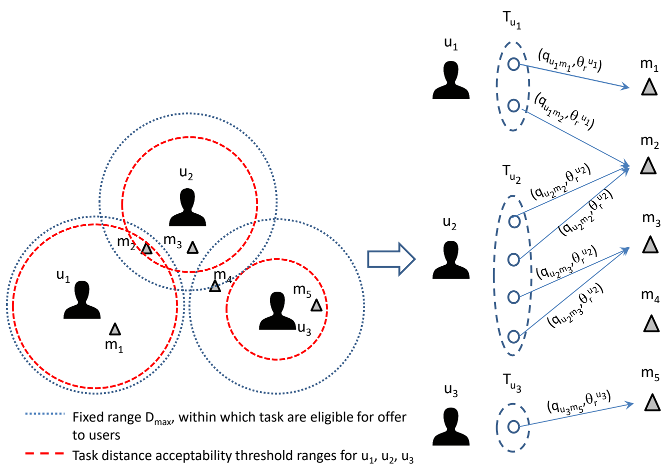



Fig. 2. Paired task offers, toy example with $|\mathcal{U}|$=3, $|\mathcal{M}|$=5. Users are assumed to belong to class $U_d$ and we show the tuple sets $\mathcal{T}_u$ for each of them together with the resulting contribution quality and cost, in line with Table III. User 2 may end up contributing to task $m_2$ when getting paired offers for $(m_2, m_3)$ or $(m_2, m_4)$; user 5 may only end up contributing to task $m_5$ or not at all; and $m_4$ can no way receive a contribution since it lies beyond the distance acceptability threshold of all three users.

to be made to user for this contribution. The app will need to choose one of these tuples to offer to each user, respecting the budget constraints of each task, and aiming at maximizing the aggregate quality of user contributions to tasks.

Formally, if $x_{ut} = 1,\ t \in \mathcal{T}_u$, when a particular tuple is offered to user $u$, and $x_{ut} = 0$, otherwise, the optimization problem faced by the app is

$$\max_{\mathbf{x}} \quad \sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{T}_u} q_{um(t)} x_{ut} \tag{8}$$

$$s.t. \quad \sum_{u \in \mathcal{U}} \sum_{t \in \mathcal{T}_u : m(t) = m} r_{um(t)} x_{ut} \leq B_m \quad \forall m \in \mathcal{M} \tag{9}$$

$$\sum_{t \in \mathcal{T}_u} x_{ut} = 1 \quad \forall u \in \mathcal{U} \qquad (P2) \tag{10}$$

$$x_{ut} \in \{0, 1\}, \quad u \in \mathcal{U}, t \in \mathcal{T}_u \tag{11}$$

The problem (P2) resembles the problem (P1) that emerged in the case of single-task offers. We could again view tasks as bins of capacity equal to the task budget. The difference is that the items are now the tuples that can be generated out of the set $\mathcal{P}_u$ for each user $u$ and there is the additional constraint that only one of these tuples can be offered for each user. Namely, when compared to the original maximum Generalized Assignment Problem (GAP), the items are grouped in classes (one per user) and only one of those can be assigned to a bin. This problem is also NP-hard since it generalizes the maximum Generalized Assignment Problem [14].

## V. EVALUATION - NUMERICAL RESULTS

In this section, we evaluate the achievable performance gain when the incentive allocation process explicitly accounts for the lexicographic decision-making strategies of end users.

### A. *Methodology*

We simulate instances of the joint task recommendation and reward offer problem for the case of single-task offers. Each problem instance specifies the number of MCS tasks, $M$, and their spatial distribution across a rectangular area of $R$x$R\ m^2$; the number of involved users, $U$, and their positions within the same area; and the decision-making profiles of the users, *i.e.*, the type of FFT driving the responses of each one to task and reward offers and the relevant thresholds, $\theta_r^u$ and $\theta_d^u$. The task and user locations, as well as the decision-making profiles of users (FFT type, threshold values) are randomly chosen, in line with different statistical distributions.

For each problem instance, we solve the optimization problem (P1) with the approximation algorithm in [15] to compute the aggregate expected quality of contributions. The approximately optimal solution (GAP) is compared to four alternative heuristics that determine the recommended tasks and offered rewards without accounting for the lexicographic structure in the user decision making. Hence, recommendations may be issued to user $u$ for the task $m$ that lies closest to her so that

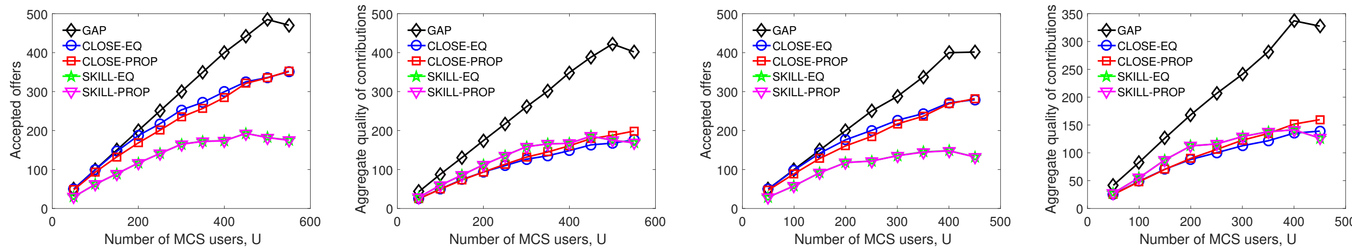
$$x_{um} = 1, \quad m = \arg\min_{l \in \mathcal{M}_u} d_{ul}, \quad u \in \mathcal{U}, \tag{12}$$

or for the one she is most skilled for, *i.e.*,

$$x_{um} = 1, \quad m = \arg\max_{l \in \mathcal{M}_u} q_{ul}, \quad u \in \mathcal{U}. \tag{13}$$

On the other hand, the task budget may be split either equally among users that get recommendations for it, that is

$$r_{um} = \frac{B_m}{K}, \quad K = |\{u : x_{um} = 1\}| \tag{14}$$




(a) $R$ = 500, $M$ = 25, $\theta_d^u \sim unif(100, 250)$ (b) $R$ = 750, $M$ = 20, $\theta_d^u \sim unif(100, 350)$

Fig. 3. Approximately optimal task recommendations and reward offers *vs.* alternative heuristic rules: $\theta_r^u \sim unif(0.5, 1.5)$.

TABLE IV
HEURISTIC RULES FOR RECOMMENDING TASKS AND OFFERING REWARDS TO MCS USERS.

| Recommended task / Budget split | Closest | User is most skilled for |
|---|---|---|
| Equally | CLOSE-EQ | SKILL-EQ |
| In proportion to user skills | CLOSE-PROP | SKILL-PROP |

or in proportion to the quality of the contribution they can make to it (*skill*),

$$r_{um} = \frac{q_{um}}{\sum\limits_{v:x_{vm}=1} q_{vm}} B_m \qquad (15)$$

The combinations of the two task recommendation rules (12)-(13) with the reward offer rules (14)-(15) result in four heuristics for the joint task recommendation and reward offer problem, which are hereafter abbreviated as CLOSE-EQ, CLOSE-PROP, SKILL-EQ, and SKILL-PROP and are summarized in Table IV. We also introduce the abbreviations CLOSE-* and SKILL-* as pointers to the first two and last two heuristics, respectively. The task recommendations and reward offers determined by each of those four heuristics are then processed by the decision trees profiling each user to determine whether the user will accept the offer and contribute to the task or not.

### B. Numerical results

*1) General performance trends:* Figure 3 plots the number of accepted offers and the resulting cumulative quality of user contributions to MCS tasks as the density of users per MCS task grows up to more than 20 ($U/M > 20$). In these experiments, the users are equally split between the four FFT types in Fig. 1. Four remarks are worth making in Fig. 3.

First, GAP significantly outperforms the four heuristics. It manages the budget allocated to each task more efficiently since it has the flexibility to recommend any task to a user and adapt the offered reward to her specific decision-making process (see Eq. (3)). On the contrary, the heuristics issue fixed and suboptimal task recommendations to each user since they are not aware of their satisfying thresholds. The performance gap grows as the user density increases up to a point that the available budget does not suffice to reward candidate users and GAP cannot ensure acceptance of all task offers.

Secondly, focusing on the four heuristics, the task recommendation rule weighs more than the reward allocation rule. The latter has practically no impact when users get recommendations for tasks they are most skilled for. All users who share the task budget in this case feature high values of $q_{um}$ so that sharing the budget in proportion to these values (SKILL-PROP) is almost identical to sharing it equally among users (SKILL-EQ). On the other hand, the two CLOSE-* heuristics attract more contributors, especially at low user densities, but do not capitalize this feature in terms of quality of contributions. Targeting the reward offers to more skilled users in SKILL-* more than compensates for the fewer attracted user contributions.

Thirdly, this last trend generalizes to the comparison of the two task recommendation rules. The attracted quality of contributions is higher under the SKILL-* heuristics, even if they end up attracting considerably fewer contributors than the CLOSE-* heuristics. Finally, the users who decline offers under the heuristic rules or under GAP at high user density, are almost exclusively those who decide according to the strict pectinate models (ref. Fig. 1); for those users both thresholds, $\theta_d^u$ and $\theta_r^u$, need to be satisfied. This is also the reason why the CLOSE-EQ and CLOSE-PROP heuristics manage to involve more contributors. By recommending the tasks that lie closest to the user, they (almost always) satisfy directly the first threshold and only fail to get contributions from users who cannot get a satisfying reward ($r_{um} < \theta_r^u$). We look more closely how the mix of strict *vs.* lenient pectinates affects performance in the following paragraph.

*2) Impact of decision-making model mix:* Users who decide in line with a strict pectinate FFT (cases b. and d. in Fig. 1) are more demanding in that they decline an offer if either cue fails to satisfy the respective threshold. As a result, the platform has less flexibility in handling them.

Fig. 4 suggests that the performance of all algorithms is hurt as the portion of these users increases. The performance deterioration is most dramatic for the two SKILL-* heuristics at both low and high user density. The two distance-oriented heuristics are more robust to the variation of the user mix but perform worse than all the alternatives already in the absence of strict pectinate users.

The impact on GAP has different qualitative features at low and high user density. For $U$ = 200 users, GAP can still

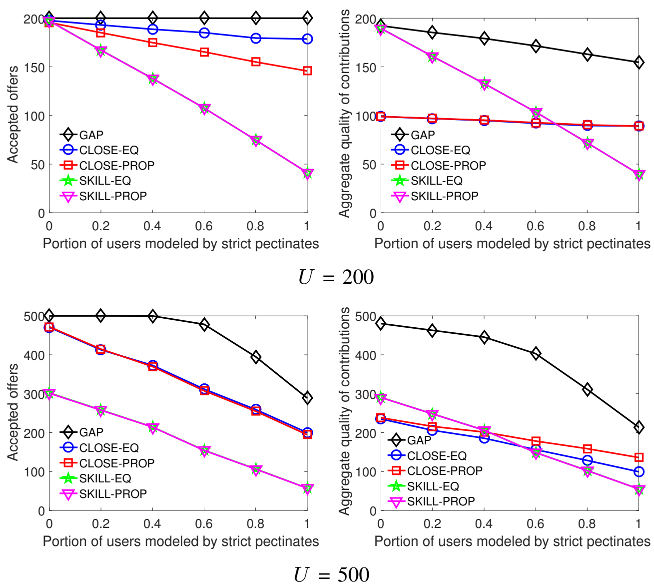



Fig. 4. Strict *vs.* lenient pectinates: $R = 500$, $M = 25$, $\theta_r^u \sim unif(0.5, 1.5)$, $\theta_d^u \sim unif(100, 500)$.

tailor its task recommendations and reward offers so that all of them get accepted. Yet, it compromises with respect to the quality of attracted contributions. As the number of strict pectinate users increases, the users are increasingly making contributions to tasks they are less skilled for. For $U = 500$ users, GAP also loses its capacity to ensure task contributors. The quality of attracted contributions drops more dramatically, but it still remains better than its alternatives.

*3) Impact of reward threshold:* Higher spread of the reward threshold values $\theta_r^u$ implies more users with higher reward demands in order to contribute to a task. Figure 5 plots the aggregate quality of user contributions against increasing values of $r_{max}$, the maximum value of $\theta_r^u$ over all users, under different user densities. The users, as with Fig. 3 are equally split between the different FFT types in Fig. 1.

GAP confirms the performance advantage evidenced in the earlier experiments, experiencing a performance drop with higher values of $r_{max}$ only when $U = 500$. The quality of contributions is higher under SKILL-* than with CLOSE-*, but the gap closes at higher $r_{max}$ and user density values. After some "critical" $r_{max}$ value, which moves leftwards on x-axis as $U$ increases, the better quality of attracted contributions under SKILL-* cannot compensate for their shrinking number and the two heuristics fall behind the CLOSE-* heuristics.

## VI. Related Work

There exists a large body of research work on incentives for mobile crowdsensing. Part of this work explicitly considers the quality of user contributions, for example [17] [18] [19]. In [17], a market framework is considered that involves data contributors, service consumers, and a service provider. A novel metric is introduced to measure the quality and timeliness of contributed data and to determine the compensation of contributors and service consumption rate of service consumers. In [18], a payment scheme is devised that pays participants in accordance with their effective contribution. The anticipated quality of contributed information by each user is estimated based on the mutual information principle and the overall quality of gathered data through an expectation maximization algorithm. In [19], the aggregate quality of all tasks minus the set of costs that users undergo, is maximized through a truthful incentive mechanism based on reverse combinatorial auctions. Potential data contributors place bids on subsets of tasks they may contribute data to. The mechanism takes into account the envisioned qualities and outputs the selected winners and their compensations for executing the declared subsets of tasks. In [20], the optimal auction framework is used to design a data market that takes into account the strategic behavior of data contributors, who may misreport the cost of data contributions. An incentive-compatible mechanism determines participation levels and payments to users with the aim to minimize data acquisition cost and ensure a certain quality of aggregate information. In [21] the objective is to select a subset of data contributors for maximizing the total utility minus the sum of payments. The submodularity of the objective is exploited to devise a truthful greedy algorithm and show its effectiveness.

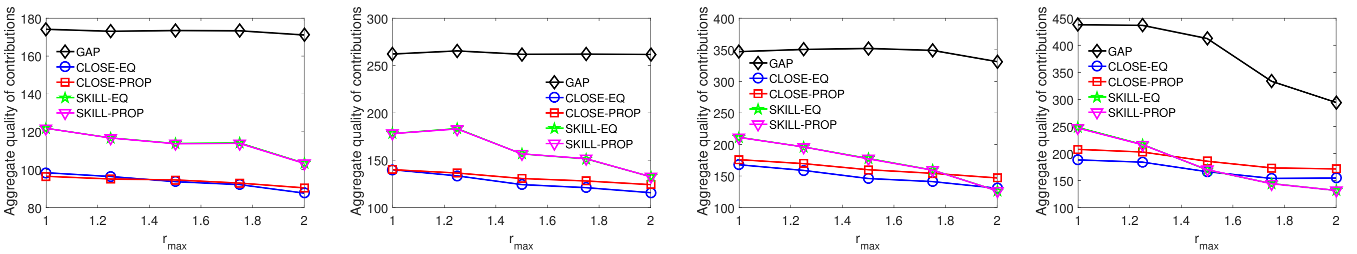
Fewer studies have devoted effort to learning user preferences and infer decision-making processes, either through online questionnaires [22] or combining real data from social networks [23]. These studies come up with probabilistic models for the dependence of users' contributions on the incentives offered to them and manage the MCS task budgets so that the total expected quality of user contributions is maximized.

Nevertheless, in all these works, users are approached as fully rational agents that seek to maximize a given (closed-form) utility function; none of them accounts for the well reported bounded rationality of human choices [3]. In this work, we tried to address this gap through interpreting modeling constructs from the field of cognitive psychology within the optimization framework of MCS campaigns.

## VII. Conclusions and Discussion

Concepts from behavioral science remain, to the best of our understanding, largely unexploited by the wireless networking community. In this work, we have made a first attempt to accommodate the extensive experimental evidence on the bounded rationality of human-decision making in the problem of MCS task recommendation and incentive allocation. We have drawn on two models for the user decision-making process from the field of cognitive psychology (fast-and-frugal trees and DEBA) and have shown how rewards can be tuned to optimize the aggregate quality of contributions to MCS tasks.

Note that we have *assumed* the relevance of FFTs and DEBA in the MCS setting, rather than *testing* it; this would demand statistically large datasets from real MCS applications, which are missing in literature. On the other hand, an attractive feature of cognitive heuristics as a whole, is that they constitute *descriptive* decision-making models. Namely, they capture salient cognitive processes (parsing of alternatives, elimination, satisficing) underlying human decisions/choices




(a) $U = 200$ (b) $U = 300$ (c) $U = 400$ (d) $U = 500$

Fig. 5. Sensitivity of aggregate quality of task contributions to the user reward threshold: $R$=500, $M$=25, $\theta_r^u \sim unif(0.5, r_{max})$, $\theta_d^u \sim unif(100, 250)$.

that persist across very different choice settings. The decision environment of the MCS user will likely be characterized by: a) limited attention and interference due to competition from other simultaneous tasks (multi-tasking and high cognitive load); b) limited time to evaluate the proposed tasks and arrive at a decision (time-pressure). Cognitive load and time pressure are both well known to favor the use of heuristics, such as the ones we have employed here, over more complex normative models of human decision-making [5] [24].

On a more general note, our work appears to have implications for a broader set of application areas beyond mobile crowdsensing, in which end-users make choices through the mediation of mobile platforms. For example, in smart-energy apps, energy-saving recommendations and consumption plans are issued to the user through the app with the goal to optimize energy savings. In mobile advertising, ads or offers are projected to users, and the aim is to optimize revenue through user response to ads. In all cases, the goal is to best *engineer* the alternatives offered to users, possibly by exploiting recommender systems' practices, and tailor incentives to user preferences so as to nudge them towards desirable choices for the user experience and the platform welfare.

## Acknowledgment

The research work of M. Karaliopoulos received funding from the Hellenic Foundation for Research and Innovation (HFRI) and the Hellenic General Secretariat for Research and Technology (GSRT), under grant agreement No 892. I. Koutsopoulos acknowledges the support from the AUEB-internal project "Original scientific publications".

## References

[1] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, Nov. 2011.
[2] P. Micholia, M. Karaliopoulos, I. Koutsopoulos, L. M. Aiello, G. D. F. Morales, and D. Quercia, "Incentivizing social media users for mobile crowdsourcing," *Int. J. Hum.-Comput. Stud.*, vol. 102, no. C, pp. 4–13, Jun. 2017.
[3] T. Luo, S. S. Kanhere, J. Huang, S. K. Das, and F. Wu, "Sustainable incentives for mobile crowdsensing: Auctions, lotteries, and trust and reputation systems," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 68–74, March 2017.
[4] H. A. Simon, "Rational choice and the structure of the environment," *Psychological Review*, vol. 63, no. 2, pp. 129–138, 1956.
[5] J. W. Payne, E. Johnson, and J. Bettman, *The Adaptive Decision Maker*. Cambridge, England: Cambridge University Press, 1993.
[6] D. Kahneman, *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux, 2011.
[7] G. Gigerenzer and D. G. Goldstein, "Reasoning the Fast and Frugal Way: Models of Bounded Rationality," *Psychological Review*, vol. 103, no. 4, pp. 650–669, 1996.
[8] A. Dieckmann, K. Dippold, and H. Dietrich, "Compensatory versus noncompensatory models for predicting consumer preferences," *Judgment and Decision Making*, vol. 4, no. 3, pp. 200–213, April 2009.
[9] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990.
[10] K. V. Katsikopoulos, "Psychological heuristics for making inferences: Definition, performance, and the emerging theory and practice," *Decision Analysis*, vol. 8, no. 1, pp. 10–29, Mar. 2011.
[11] L. Martignon, O. Vitouch, M. Takezawa, and M. R. Forster, "Naive and yet enlightened: From natural frequencies to fast and frugal decision trees," *Thinking: Psychological Perspectives on Reasoning, Judgment and Decision Making, Ch. 10*, pp. 189–211, 2005.
[12] L. Martignon, K. V. Katsikopoulos, and J. K. Woike, "Categorization with limited resources: A family of simple heuristics," *Journal of Mathematical Psychology*, vol. 52, no. 6, pp. 352 – 361, 2008.
[13] N. D. Phillips, H. Neth, J. K. Woike, and W. Gaissmaier, "FFTrees: A toolbox to create, visualize, and evaluate fast-and-frugal decision trees," *Judgment and Decision Making*, vol. 12, no. 4, pp. 344–368, 2017.
[14] D. B. Shmoys and E. Tardos, "Approximation algorithm for the generalized assignment problem," *Mathematical Programming, Series B*, vol. 62, no. 3-8, pp. 461–474, 1993.
[15] R. Cohen, L. Katzir, and D. Raz, "An efficient approximation for the generalized assignment problem," *Information Processing Letters*, vol. 100, no. 4, pp. 162 – 166, 2006.
[16] L. Fleischer, M. X. Goemans, V. S. Mirrokni, and M. Sviridenko, "Tight approximation algorithms for maximum general assignment problems," in *Proc. 17th ACM-SIAM SODA*, 2006, pp. 611–620.
[17] C.-K. Tham and T. Luo, "Quality of contributed service and market equilibrium for participatory sensing," *IEEE Trans. on Mobile Computing*, vol. 14, no. 4, pp. 829–842, April 2015.
[18] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proc. 16th ACM Mobihoc*, Hangzhou, China, June 2015, pp. 177–186.
[19] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proc. 16th ACM Mobihoc*, Hangzhou, China, 2015.
[20] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *Proc. IEEE INFOCOM '13*, Turin, Italy, April 2013, pp. 1402–1410.
[21] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. 18th ACM Mobicom*, Istanbul, Turkey, 2012, pp. 173–184.
[22] M. Karaliopoulos, I. Koutsopoulos, and M. Titsias, "First learn then earn: optimizing mobile crowdsensing campaigns through data-driven user profiling," in *Proc. 17th ACM MobiHoc*, Paderborn, Germany, July 2016, pp. 271–280.
[23] P. Micholia, M. Karaliopoulos, and I. Koutsopoulos, "Mobile crowdsensing incentives under participation uncertainty," in *Proc. 17th ACM MobiHoc MSCC workshop*, Paderborn, Germany, July 2016.
[24] L. Spiliopoulos and A. Ortmann, "The bcd of response time analysis in experimental economics," *Experimental Economics*, vol. 21, no. 2, pp. 383–433, Jun 2018.