Behavioral Factors in Market User Interface Design *

Sven Seuken[†], David C. Parkes[‡], Eric Horvitz[§], Kamal Jain[¶], Mary Czerwinski[§] and Desney Tan[§]

March 30, 2013

Abstract

Despite the pervasiveness of electronic markets in our lives, only little is known about the role of user interfaces (UIs) in promoting good performance in market domains. How does the way we display market information to end-users, and the set of choices we offer, influence economic efficiency? In this paper, we introduce a new research agenda on "market user interface design." We take the domain of 3G bandwidth allocation as an illustrative example, and consider the design space of UIs in terms of varying the number of choices offered, fixed vs. changing market prices, and situation-dependent choice sets. The UI design induces a Markov decision process, the solution to which provides a gold standard against which user behavior is studied. We provide a systematic, empirical study of the effect of different UI design levers on user behavior and market performance, along with considerations of behavioral factors including loss aversion and position effects. Finally, we fit a quantal-best response model to users' actions and evaluate a behaviorally-optimized market user interface.

Keywords: Market Design, UI Design, Behavioral Economics, Experiments.

*We we would like to thank Alvin E. Roth, Alain Cohn, Haoqi Zhang, Yiling Chen, David Laibson, and the anonymous reviewers for their helpful comments on this work. Furthermore, we are thankful to the feedback we received from various seminar/conference participants at the following institutions/events (in chronological order): Microsoft Research, Harvard University, University of Zurich, Yahoo! Research, University of Michigan, AMMA'11, the Aarhus Workshop on New Trends in Mechanism Design, the Toulouse Workshop on the Psychology and Economics of Scarce Attention, Brown University, the NBER Market Design Working Group Meeting, EC'12, the 1st International Workshop on Experimental Economics and Finance, and the University of Freiburg. Part of this work was done while Seuken was an intern at Microsoft Research. Seuken gratefully acknowledges the support of a Microsoft Research PhD Fellowship during that time. An extended abstract of this paper has previously appeared in the *Proceedings of the Second Conference on Auctions, Market Mechanisms and Their Applications (AMMA)*, New York, NY, August 2011. Furthermore, a shorter version of this paper has previously appeared in the *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, Valencia, Spain, June 2012.
[†]Department of Informatics, University of Zurich, Binzmühlestraße 14, 8050 Zürich, seuken@ifi.uzh.ch
[‡]School of Engineering & Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, parkes@eecs.harvard.edu.
[§]Microsoft Research, One Microsoft Way, Redmond, WA 98052, {horvitz, marycz, desney}@microsoft.com
[¶]eBay Research Labs, 2065 Hamilton Avenue, San Jose, CA 95125, kamaljain@gmail.com

1. Introduction

Electronic markets are becoming more and more pervasive but a remaining research challenge is to develop user interfaces (UIs) to promote effective outcomes for users. This is important because markets often present users with a very large number of choices, making it difficult for users to find the optimal choice. For example, the markets for digital content which we can access via Amazon or iTunes are growing exponentially in size. Soon, we will have to deal with many complex markets in unfamiliar domains, and react to more frequent price changes. The smart grid domain is a prime example for such a domain. As we are asked to make market decisions more and more frequently, deliberation gets costly and we cannot spend too much time on individual decisions. This is where Herb Simon's 40-year old quote still says it best:

> *"...a wealth of information creates a poverty of attention..."*
>
> Herbert A. Simon (1971), pp. 40–41.

Because humans incur cognitive costs when processing information (Miller, 1956), a wealth of information, or a wealth of choices in market environments makes attention a scarce resource. Yet, traditional economic models assume agents to be perfectly rational, with unlimited time and unbounded computational resources for deliberation. We address this discrepancy by explicitly taking behavioral considerations into account when designing market UIs. The same way that color-coded planes make the job of an air-traffic controller easier, it is our goal to design market UIs that make economic decision making easier, thereby improving social welfare. In this sense, our approach is in line with the "choice architecture" idea put forward by Thaler, Sunstein and Balz (2010).

A market UI can best be defined via two questions: first, what information is displayed to the user? Second, how many and which choices are offered to the user? Our ultimate goal is to develop a computational method that finds the optimal market UI, given a behavioral user model. Using behavioral models may lead to different market UIs for multiple reasons. For example, taking into account that users make mistakes, it may be best not to offer some choices that can lead to particularly bad outcomes (e.g., spending too much of your budget in one step). So far, the market design literature has largely ignored the intersection of market design and UI design. We argue that this intersection is important because the complexity of the UI defines the cognitive load imposed on users. Furthermore, the UI defines how, and how well, users can express their preferences. The design of good market UIs is particularly important for *electronic markets*, where individual decisions can often have low value (e.g., buying an article online for \$0.50, or reducing your room temperature by $1°C$), but the repeated decisions of millions of people still have a large effect. Furthermore, electronic markets also offer an unparalleled opportunity for market UI design. In such market domains we often haven a wealth of information about the individual market participants, and we can personalize market UIs, tailoring them to the individual user's needs and preferences. Thus, when designing an electronic market, the design of the market's UI may be as important as the market's economic design.

1.1. Overview of Results

This paper introduces a new research agenda on "market user interface design." We first present a systematic, empirical exploration of the effect that different UI designs have on users' performance in economic decision making. Then we study the automatic optimization of market UIs based on a behavioral quantal response model. We situate our study in a hypothetical market for 3G bandwidth where users can select the desired speed level, given different prices and values. While there is a possibly infinite set of choices (possible speed levels), the market UI only exposes some finite number. As the market UI designer, we get to decide how many and which choices to offer.

The participants of our experiments play a series of single-user games, facing a sequential decision-making problem with inter-temporal budget constraints. We vary a) the number of choices offered to the users (3, 4, 5, or 6), b) whether prices are fixed or dynamic, and c) whether choice sets are fixed or adaptive. Additionally, we also learn a quantal response model based on parts of the experimental data, and use computation to *automatically* optimize the market UI given the behavioral model. We then compare the behaviorally-optimized UI with a standard UI. Because the market UI has a finite number of choices, the optimization algorithm must make a trade-off between having some choices at the lower end of the speed levels (which may be the best choice when values are low and prices are high) and some choices at the upper end (which may be the best choice when values are high and prices are low). Our main results are:

1. Users' realized value increases as we increase the number of choices from 3 to 4 to 5, with no statistically significant difference between 5 and 6 choices.
2. The realized value is higher with adaptive choice sets compared to fixed choice sets.
3. The realized value is *lower* when using the UI that is optimized for behavioral play, compared to the UI that is optimized for perfectly-rational play.

The third result was particularly surprising and prompted a more detailed analysis of users' decisions. We find that our user model, based on the quantal response model, was too simplistic, with significant negative consequences for the market UI design. Our analysis suggests that we omitted important behavioral factors like *loss aversion* and *position effects*. Furthermore, we identify large differences between individual users' *level of rationality*. We find that for the "less rational" users there was no statistically significant difference in realized value using the UI optimized for rational play or optimized for behavioral play. However, the more rational users suffered, because the UI optimization took away too many of the valuable choices, making the decision problem easier, but reducing the total realized value. Thus, this result points towards the need for personalized market UIs that take into account each user's individual level of rationality.

1.2. Related Work

Prior research has identified a series of behavioral effects in users' decision making. Buscher, Dumais and Cutrell (2010) show that the amount of *visual attention* users spend on different parts of a web page depends on the task type and the quality of the information provided. Dumais, Buscher and Cutrell (2010) show that these "gaze patterns" differ significantly from user to user, suggesting that different UIs may be optimal for different groups of users. In a study of the cognitive costs associated with decision making, Chabris et al. (2009) show that users allocate time for a decision-making task according to cost-benefit principles. Because time is costly, more complex UIs put additional costs on users.

In addition to UI complexity, emotional factors are also important in decision making. Consider the "jam experiment" by Iyengar and Lepper (2000), which shows that customers are happier with the choices they make when offered 6 different flavors of jam compared to 24 different flavors. Schwartz (2005) identifies multiple reasons why more choices can lead to decreased satisfaction, including *regret*, *missed opportunities*, *the curse of high expectations*, and *self blame*. While emotional factors are important in many domains, in this paper we do not aim to study them directly. Instead we focus on users' cognitive limitations and corresponding bounded rationality.

Some research on UIs for recommender systems addresses aspects related to our work. Knijnenburg et al. (2012) study which factors explain the user experience of recommender systems. Chen and Pu (2010) propose methods for dynamically changing a recommender system UI based on user feedback, to help users find suitable products in very large domains. Hauser et al. (2009) present a completely automated approach for dynamically adapting user interfaces for virtual advisory websites. They are able to infer users' cognitive styles based on click-stream data and then adjust the look and feel of a website accordingly. However, in contrast to our work, their goal is to increase users' purchase intentions, while our goal is to help users make better decisions.

Horvitz and Barry (1995) present a framework for the design of human-computer interfaces for time-critical applications in non-market-based domains. Their methodology trades off the costs of cognitive burden with the benefits of added information. Johnson, Payne and Bettman (1988) show that the way information is displayed (e.g., fractional vs. decimal probability values) has an impact on user decision making. The authors briefly discuss the implications of their findings for the design of information displays.

The work most closely related to ours is SUPPLE, introduced by Gajos, Weld and Wobbrock (2010), who present a system that can *automatically* generate user interfaces that are adapted to a person's devices, tasks, preferences, and abilities. They formulate the UI generation as an optimization problem and find that automatically-generated UIs can lead to significantly better performance compared to manufacturers' defaults. While their approach is in line with our goal of "automatic UI optimization," they do not consider a market context. They build a model of users' pointing and dragging performance and optimize their UIs for accuracy, speed of use, and users' subjective preferences for UI layouts. In contrast, we build a *behavioral user model* and optimize for *decision quality* in market environments where users are dealing with values, prices, and budgets.

In our own prior work (Seuken, Parkes and Jain, 2010), we have introduced the goal of designing "hidden markets" with *simple* and *easy-to-use* interfaces. In related work (Seuken et al., 2010*a*,*b*), we have presented a UI for a P2P backup market, demonstrating that it is possible to hide many of the market's complexities, while maintaining a market's efficiency. Similarly, Teschner and Weinhardt (2011) show that users of a prediction market make better trades when using a simplified market interface, compared to one that provides the maximum amount of information and trading options. This paper is in the same vein as this prior work, but presents the first *systematic* exploration of the market UI design space, thereby opening up a new field of empirical research.

2. The Bandwidth Allocation Game

The experiments described in this paper were conducted as part of a larger user study on people's experiences and preferences regarding smartphone usage.[1] The international smartphone market is a Billion-dollar market with more than 100 Million users worldwide. With an ever growing set of bandwidth-hungry applications on these phones, the efficient allocation of 3G or 4G bandwidth is an important problem. According to Rysavy Research (2010), the demand for wireless bandwidth will continue to grow exponentially over the next few years and it will be infeasible for the network operators to update their infrastructure fast enough to satisfy future demand. The common approach for addressing the problem of bandwidth demand temporarily exceeding supply is to slow down every user in the network and to impose fixed data usage constraints. Obviously, this introduces large economic inefficiencies because different users have different values for high speed vs. low speed internet access at different points in time.

Now imagine a hypothetical market-based solution to the bandwidth problem. The main premise is that users sometimes do tasks of high importance (e.g., send an email attachment to their boss) and sometimes of low importance (e.g., random browsing). If we assume that users are willing to accept low performance now for high performance later, then we can optimize the allocation of bandwidth by shifting excess demand to times of excess supply. Figure 1 (a) shows a mock-up application for such a bandwidth market. Imagine that at the beginning of the month each users gets 50 points, or tokens. As long as there is more supply than demand, a user doesn't need to spend his tokens.[2] However, when there is excess demand and the user wants to go online, then a screen pops up (as shown in Figure 1 (a)), requiring the user to make a choice. Each speed level has a different price (in tokens). For simplicity, we assume that when a user runs out of tokens, he gets no access or some

[1] While future experiments will show how our results translate to other domains, it is important to note that the four design levers we study constitute *within-experiment* variations. Thus, any *change* in behavior can be attributed to changes in the UI and are likely not specific to this domain.

[2] Throughout this paper, when referring to "users", we always mean male as well as female users, but to simplify the language, we only use "he" and "his."

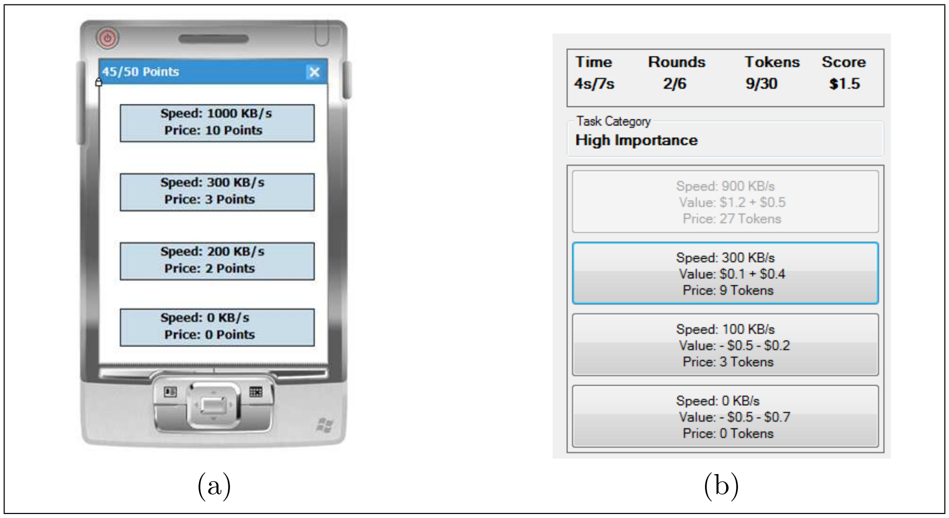

Figure 1: (a) Mockup of the bandwidth market UI. (b) Screenshot of the market game used in the experiments.

very slow connection.[3] This domain is particularly suitable to studying market UIs because we can easily change many parameters of the UI, including the number of choices, whether prices stay fixed or keep changing, and the particular composition of the choice set.

2.1. Game Design

Figure 1 (b) shows a screenshot of the market game we designed for our experiments, mirroring the mockup of the market application, except that now the value for each choice is no longer private to each user, but explicitly given to the user. Note that this is a single-user game on top of a *simulated market domain*. Each game has 6 rounds, and the user has a fixed budget of 30 tokens that he can spend over the course of the 6 rounds. In addition to the current round and budget, the state of the game is determined by the current set of *values* and associated *prices* of all available choices, and those change stochastically from round to round.

In each round of the game, the user has to select one of the choices. Each choice (i.e., a button in Figure 1 (b)) has three lines: the first line shows the *speed* of that choice in KB/s. The second line shows the *value* of that choice in dollars (as a sum of two values). The value represents the dollar amount that is added to a user's *score* when that choice

[3]In this paper, we do not concern ourselves with different business models or market designs for this domain. In particular, we do not address the question whether users should be allowed to pay money to buy more tokens. We do not suggest that this particular business model of using a fixed number of tokens per month should be adopted. Instead, we merely use this hypothetical market application as a motivating domain for our experiments into market UI design.

is selected. The third line shows the *price* of that choice in tokens. Note that to play the game optimally, the user only needs to know the values and the prices of each choice, but not the *speeds*. However, we also include the *speed* information to label the buttons such that it is easier for users to recognize what has changed in the current round (e.g., values and/or prices). When the user selects a particular choice, the corresponding number of tokens is subtracted from his budget and the corresponding value is added to his *score* which is displayed in the top right corner of the window. The score after the 6th round is the final score for the game.

Next to the score is a label displaying the user's current budget, which always starts at 30 in round 1 and then goes down as the user spends tokens. With the user's budget decreasing during a game, choices that have a price higher than the user's current budget become unavailable and are greyed out (as is the case for the top choice in Figure 1 (b)). To the left of the user's budget, the game shows the number of rounds that are left until the game is over. Finally, at the very left of the window, we show the user how much time he has left to make a decision in this particular round (e.g., in Figure 1 (b) the user still has 4 seconds left to make a decision in the current round). We put users under time pressure to induce a certain error rate that allows for a meaningful comparison of different market UIs. To have a reasonably high error rate even for easy games (e.g., with just three choices), we display the value of each choice as a sum of two values. This induces extra cognitive effort and thus leads to a higher error rate.

In every round, the user is in one of three task categories (high importance, medium importance, and low importance), which is displayed in the *task category label*. Every round, one of these three categories is chosen randomly with probability 1/3. Note that this corresponds to the original premise that users are doing tasks of different importance at different points in time. The task category determines the values of all choices. Effectively, the user has three *concave value functions* that map bandwidth levels to values. Table 1 shows an overview of the values the user can expect in the three categories for a game with 4 choices. As one would expect, selecting the higher speed choices in the "high importance" category gives the user a very high value, while choosing low speeds in the high importance category leads to a severe penalty. In contrast, in the "low importance" category the user can earn less value for selecting high speeds, but is also penalized less for selecting the lowest speed. However, the values shown in Table 1 are only the averages of the values in each category. In every round, the actual value for each choice is perturbed upwards or downwards with probability 1/3, to introduce additional stochasticity in the game. This avoids that users can memorize a fixed set of values for each task category, and also rules out that simple heuristics led to near-optimal game play.

The user's problem when playing the game is to allocate the budget of 30 tokens optimally over 6 rounds, not knowing which values he will face in the future. For some experimental treatments, we also randomly vary the prices charged for each of the choices from round to round. Thus, the user may also have uncertainty about which price level he will be facing next. This problem constitutes a sequential decision making problem under uncertainty.

	High Imp.	Medium Imp.	Low Imp
900 KB/s	\$1.7	\$1.1	\$0.4
300 KB/s	\$0.5	\$0.2	- \$0.2
100 KB/s	-\$0.3	- \$0.3	- \$0.5
0 KB/s	- \$1	- \$0.9	- \$0.8

Table 1: The Values in the 3 different Task Categories

We calibrated the game (i.e., the size of the budget, the nominal values of the choices, the prices, the number of rounds, etc.) in such a way that random play has a highly negative expected score, but such that optimal play leads to an average score between \$0.50 and \$1.30 (depending on the particular experimental treatment). Thus, to play the game well and achieve positive scores, the users had to exert significant cognitive effort and properly take the multi-step stochastic nature of the game into account.

2.2. MDP Formulation and Q-Values

Each game can formally be described as a finite-horizon Markov Decision Problem (MDP) without discounting (Puterman, 1994):

- **State Space:** *CurrentRound* × *CurrentBudget* × *CurrentCategory* × *CurrentValueVariation* × *CurrentPriceLevel*.
- **Actions:** Each choice affordable in the current round given current budget.
- **Reward Function:** The value of each choice.
- **State Transition:** The variables *CurrentRound*, *CurrentBudget*, and *CurrentScore* transition deterministically given the selected choice. The other variables *CurrentCategory*, *CurrentValueVariation* and *CurrentPriceLevel* transition stochastically.

The largest games we consider have approximately 7 million state-action pairs. Using dynamic programming, we can solve games of this size quickly (in less than 20 seconds). Thus, we can compute the optimal MDP policy, and we always know exactly which choice is best for each possible situation (game state) that can arise. Note that this policy is, of course, computed assuming that the future states are not known; only the model and the transition probabilities as described above are known.

Solving the MDP involves the computation of the *Q-values* for each state-action pair. For every state s and action a, the Q-value $Q(s, a)$ denotes the expected value for taking action a in state s, and following the optimal MDP policy for every subsequent round. Thus, the optimal action in each state is the action with the highest Q-value, and by comparing the differences between the Q-values of two actions, we have a measure of how much "worse in expectation" an action is, compared to the optimal action.

2.3. The Quantal-Response Model

A well-known theory from behavioral economics asserts that agents are more likely to make errors the smaller the cost for making that error. This can be modeled formally with the *quantal response model* (McKelvey and Palfrey (1995)) which predicts the likelihood that a user chooses action a_i in state s to be:

$$P(a_i \mid s) = \frac{e^{\lambda \cdot Q(s,a_i)}}{\sum_{j=0}^{n-1} e^{\lambda \cdot Q(s,a_j)}}$$

where n denotes the total number of actions, $Q(s, a_i)$ denotes the Q-value of action a_i in state s, and $\lambda \geq 0$ is a precision parameter indicating how sensitive users are to differences between Q-values. $\lambda = 0$ corresponds to random action selection, and $\lambda = \infty$ corresponds to perfectly-rational action selection, i.e., always choosing the optimal action. Based on experimental results, one can compute a maximum-likelihood estimate for λ, i.e., maximizing the likelihood of the observed data. Equipped with a particular λ, this constitutes a user model which we use to optimize the UI for behavioral play (see Wright and Leyton-Brown (2010) for a comparison of behavioral models).

3. Experiment Design

Before we discuss the experiment design, let's briefly pause to understand what exactly is within the control of the market UI designer, and what is not. Remember that in theory, there is an infinite set of choices (possible speed levels), but we assume that any market UI can only expose a small, finite number of choices to the user. The UI designer decides 1) how many choices and 2) which exact choices to offer. For example, as in Figure 1 (b), we can provide 4 choices, i.e., 0 KB/s, 100 KB/s, 300 KB/s, and 900 KB/s. Alternatively, we could provide 3 choices, for example 0 KB/s, 500 KB/s, and 1000 KB/s. Note that by picking the choices, we only choose the market interface; the user's *value function* which maps speed levels to values doesn't change. Of course, higher speed levels have a higher value for the user, but they also have a higher price.

In addition to the constraint of having a fixed number of choices, we also require the choice set to be fixed ex-ante and stay fixed throughout a game. In particular, the choices cannot depend on the state of the game (round, budget, category, value variation, price level). There is one exception to this rule, namely in the *Adaptive Choice Set* treatment, where we specify not one but three different UIs, one for each category *high*, *medium*, and *low* (we discuss this design lever in more detail in Section 4.2).

Thus, except in the *Adaptive Choice Set* treatment, the UI remains fixed for the 10 to 15 games that users play per treatment. For example, in the treatment with 5 choices, the user gets the same 5 choices in every round. Of course, in each of the possibly millions of different game states, a different choice is optimal. If the user could choose his speed freely, perhaps the optimal speed in some state would be 378 KB/s. But our UIs only offer a fixed, finite number of choices. Of course, despite this constraint, for every state in the game, one of the available choices is still the best, and by solving the MDP we know which one it is. But in the real world, a UI designer would also only get to pick *one* UI (possibly knowing a distribution over situations a user will be in). We as the experimenters adopt the same viewpoint: we select one fixed UI, knowing the distribution of game states that a user will encounter, but we cannot change the UI during a game.

3.1. Design Levers

We study the following four *market UI design levers*:

1. **Number of Choices:** This design lever describes how many choices (i.e., the number of buttons) are available to the users (3, 4, 5, or 6).

2. **Fixed vs. Dynamic Prices:** In the *fixed price* treatment, each choice always costs a fixed number of tokens (2 tokens per 100KB/s). With *dynamic prices*, one of 3 price levels is chosen randomly with probability 1/3 in each round, where the price per 100 KB/s is either 1, 2, or 3 tokens (thus, 500KB/s cost either 5, 10, or 15 tokens).[4]

3. **Fixed vs. Adaptive Choice Sets:** In the *fixed choice set* treatment, the users always have the same set of choices available to them in every round (e.g., always 0 KB/s, 100 KB/s, 300 KB/s, and 900KB/s). In the *adaptive choice set* treatment, the decision within the UI design as to which choices to offer is allowed to vary with the category (e.g., in the high category, more high speed choices may be available; in the low category, more low speed choices may be available).

4. **Rational vs. Behavioral UI Optimization:** This design lever describes which method is used to determine the composition of the choice sets (i.e., fixing the number of choices, which particular speed levels are available to users). In the *Rational-Optimization* treatment, the choice sets are optimized based on the MDP model assuming perfectly rational play. In the *Behavioral-Optimization* treatment, the choice sets are optimized assuming behavioral play according to the quantal response model.

3.2. Methodology and Experimental Set-up

We recruited 53 participants (27 men, 26 women) from the Seattle area with non-technical jobs. All participants had at least a Bachelors degree and we excluded participants who majored in computer science, economics, statistics, math, or physics. They were fluent English speakers, had normal (20/20) or corrected-to-normal vision, and were all right-handed. All of them used a computer for at least 5 hours per week. Their median age

[4] The motivation for testing this design lever is that in some domains, balancing supply and demand may be possible with other means than dynamic prices. However, a detailed discussion of this idea is beyond the scope of this paper. We also don't analyze this particular design lever in this paper.

was 39, ranging from 22 to 54. None of the participants worked for the same company, but all of them had some familiarity with smartphones. We ran one participant at a time with each session lasting about 1.5 hours. The users first filled out a pre-study questionnaire (5 minutes). Then they went through a training session where the researcher first explained all the details of the game and then gave them the opportunity to play 6 training games (15 minutes). After the training was over, they participated in the experiment (55 minutes) and finally completed a post-study survey (10 minutes). The participants were compensated in two ways. First, they received a software gratuity that was independent of their performance (users could choose one item from a list of Microsoft software products). Second, they received an Amazon gift card via email with an amount equal to the total score they had achieved over the course of all games they had played. The expected score for a random game, assuming perfect play, was between \$0.50 and \$1.30, depending on the particular treatment. With random action selection, the expected score was highly negative. After each game, we showed the users their score from the last game and their accumulated score over all games played so far.[5] The final gift card amounts of the 53 users varied between \$4.60 and \$43.70, with a median amount of \$24.90.

3.3. Time Limits

To study the effect of the UI design on a user's ability to make economic decisions we need a reasonably complex decision problem, such that it is neither too easy nor too difficult for users to find the optimal decision. We achieve this by making decision time a scarce resource, as prior research has shown that users make worse decisions when under time pressure (Gabaix et al., 2006). We impose an **exogenous** time limit of 12 (7) seconds per round. If a user doesn't make a choice within this time limit, the lowest choice (with 0KB/s for 0 tokens and a highly negative value) is chosen, and the game transitions to the next round. The time resets in every round. To warn the user, the game starts beeping three seconds before the end of a round. Note that the time limits of 12 and 7 seconds respectively were carefully chosen after a series of pre-test. In those pre-tests we had found that having between 7 seconds and 12 seconds put most of the users under enough time pressure such that it was difficult for them to find the optimal choice, but still gave them enough time to process enough of the available information such that they could play the game reasonably well.

In addition to the games with a fixed time limit (7 and 12 seconds), the users also played a series of games with an **endogenous** time limit. They had 240 seconds to play many games repeatedly; once a user finished one game, there was a 15 second break, and then the next game started. Thus, the cost for spending more time on a decision was internalized by the participants. We used this time treatment to study the effect of fixed vs. dynamic prices on decision time. However, we do not discuss this aspect in this paper.

[5]Originally, we had 56 participants in our study. However, in our analysis, we excluded the data from 3 participants (2 males, 1 female) because their accumulated score was more than two standard deviations away from the mean of the accumulated score in their treatment group.

Number Of Choices \ Seconds per Game	12 seconds	7 seconds	240 seconds
3	4 $\times$	4 $\times$	1 $\times$
4	4 $\times$	4 $\times$	1 $\times$
5	4 $\times$	4 $\times$	1 $\times$
6	4 $\times$	4 $\times$	1 $\times$

Table 2: Design of Experiment 1. Each participant played between 40 and 50 games. The design lever *Number of Choices* was a within-subject factor, the design lever *Fixed vs. Dynamic Prices* was a between-subjects factor.

3.4. Treatment Variations

The study was split into two separate experiments. Experiment 1 involved 35 participants, and we tested the design levers *Number of Choices* (within-subject factor) and *Fixed vs. Dynamic Prices* (between-subject factor). Table 2 depicts the experiment design for each individual user. We randomized the order in which the users played the games with 3, 4, 5, or 6 choices. For each of those treatments, every user started with the four 12-second games, then played the four 7-second games, and then the 240-second endogenous time game. In Experiment 2 we had 18 participants and we tested the design levers *Fixed vs. Adaptive Choice Sets* and *Rational vs. Behavioral UI Optimization* (both within-subject factors). Here, all games had four choices and dynamic prices. See Table 3 for a depiction of the experiment design for each individual participant. Again, we randomized the order of the treatments.

3.5. Computational UI Optimization

For a fair comparison of different market UIs (e.g., one with 4 choices vs. one with 5 choices), we chose each of these UIs *optimally*, given the constraints imposed by the treatment. The only choice that was always included was the 0 KB/s choice (for 0 tokens). Here, "optimally" means that we selected the one fixed UI with the highest *ExpectedOptimalValue* given the MDP model (i.e., distribution of game states). To make this optimization computationally feasible, we discretized the search space, with 100KB/s being the smallest unit. Our search algorithm took as input the design parameters (e.g., 3 choices and optimized for rational play), iterated through all possible combinations of choices (i.e., all possible combinations of speed levels), solved the resulting MDP for each combination, and output the UI with the highest *ExpectedOptimalValue*. Note that the optimization algorithm automatically makes a trade-off between having some choices at the lower end of the speed levels (e.g., 200 KB/s which may be the best choice when values are low and prices are high) and some choices at the upper end of the speed levels (e.g., 900 KB/s which may be the best choice when values are high and prices are low). In particular, this means that for

Treatment Variation \ Seconds per Game	12 seconds	7 seconds	240 seconds
Fixed Choice Sets & Rational Optimization	4 ×	4 ×	1 ×
Adaptive Choice Sets & Rational Optimization	4 ×	4 ×	1 ×
Fixed Choice Sets & Behavioral Optimization	4 ×	4 ×	1 ×
Adaptive ChoiceSets & Behavioral Optimization	4 ×	4 ×	1 ×

Table 3: Design of Experiment 2. Every participant played between 40 and 50 games. Both design levers *Fixed vs. Adaptive Choice Sets* and *UI Optimization* were within-subject factors.

a particular game state, the "theoretically optimal" choice for that state (if all affordable speed levels were available) will not always be among the set of offered choices. Using this UI optimization approach, we guarantee that for every particular set of design criteria, we present the user with the best possible UI given the constraints.

3.6. Hypotheses

The larger the number of choices, the higher the expected value of the game assuming optimal play. Yet, Malhotra (1982) has shown that information overload leads to poorer decisions. We hypothesized that at first, the benefit from having more choices outweighs the additional cognitive load (H1), but that as the number of choices gets large, the added cognitive costs become the dominant factor (H2). Similarly, using *AdaptiveChoiceSets* tailors the available choices to the particular task category, which should make the decision easier for the user. On the other hand, the fact that the choices may change from round to round might also make it harder for users to find the optimal one. We hypothesized that the overall effect is positive (H3). Finally, the behavioral optimization changes the composition of the choice set. In particular, it may eliminate some choices that may be useful in some game states because the behavioral model deems them as too risky. Thus, a user might suffer without those choices, or he might benefit, because the risky choices are eliminated. We hypothesized that the overall effect is positive (H4). To summarize, our four hypotheses are:

> ***H1***: *The realized value increases as we increase the number of choices.*
>
> ***H2***: *The realized value first increases as we increase the number of choices, but ultimately decreases.*
>
> ***H3***: *The realized value is higher when using adaptive choice sets, compared to using fixed choice sets.*
>
> ***H4***: *The realized value is higher when using behavioral optimization, compared to using rational optimization.*

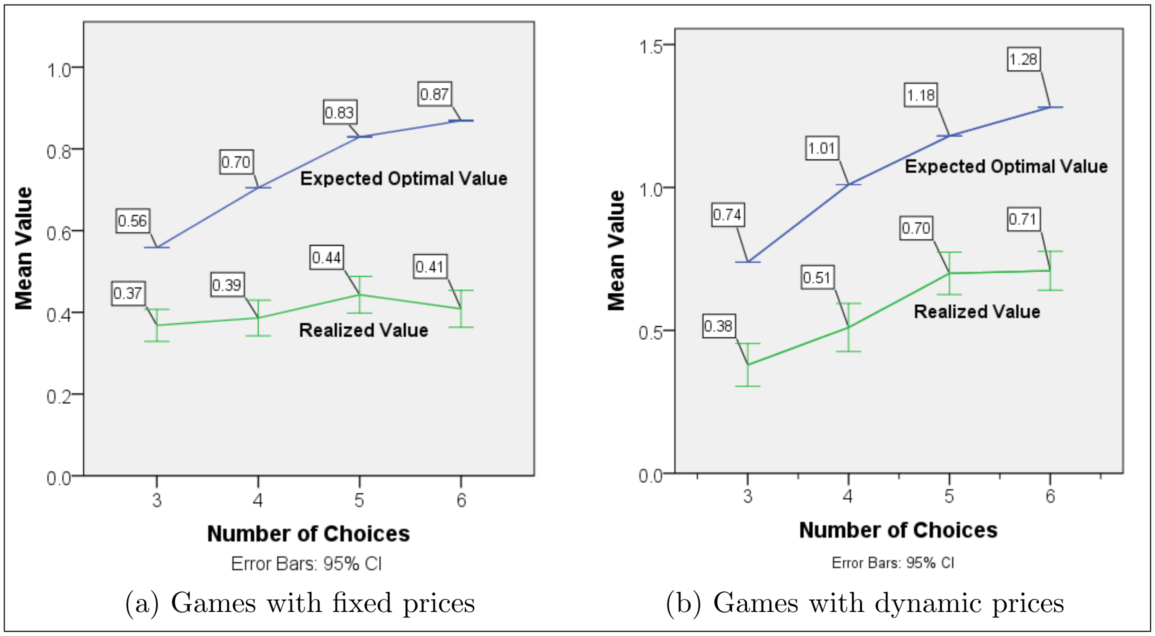

Figure 2: Mean values for 3, 4, 5, and 6 choices, in the games with (a) fixed prices, and (b) dynamic prices. The blue line (on top) corresponds to *ExpectedOptimalValue*. The green line (on the bottom) corresponds to *RealizedValue*.

4. Experimental Results

In this section, we describe the results regarding our hypotheses. As the regression technique we use *Generalized Estimating Equations (GEE)*, an extension of generalized linear models, that allow for the analysis of correlated observations (Nelder and Wedderburn, 1972). This gives us consistent coefficient estimates with robust standard errors despite using repeated measures from individual users.

4.1. Number of Choices

The first design lever we analyze is *NumberOfChoices*. We measure the effect of this design lever by analyzing the dependent variable *RealizedValue*, which is a randomness-adjusted version of the user's total score per game (see Appendix Section A for details). For this analysis, we study the games with *fixed* prices and with *dynamic* prices separately. Consider first the two graphs in Figure 2. In each graph, the blue line (on the top) represents the game's *ExptectedOptimalValue*, and the green line (at the bottom) represents the users' average *RealizedValue*. For the games with fixed prices (on the left), the graph suggests that the user's realized value first increases as we go from 3 to 4 to 5 choices, but then decreases again as we go from 5 to 6 choices (however, note the large and overlapping error bars). For the games with dynamic prices (on the right), the effects are more clear.

Factors/Covariates	(1)	(2)
Intercept	0.368**** (0.0275)	0.343**** (0.0371)
NumChoices=6	0.04 (0.0337)	0.040 (0.0331)
NumChoices=5	0.075** (0.0342)	0.077** (0.0332)
NumChoices=4	0.018 (0.0279)	0.015 (0.302)
NumChoices=3	0	0
12-SecondGame		0.003 (0.0247)
GameCounter		0.001 (0.0007)
Model Fit (QICC)	43.635	47.515

(a) Games with fixed prices

Factors/Covariates	(1)	(2)
Intercept	0.379**** (0.0650)	0.326**** (0.0859)
NumChoices=6	0.330**** (0.0740)	0.326**** (0.0725)
NumChoices=5	0.320**** (0.0609)	0.329**** (0.0610)
NumChoices=4	0.131* (0.0740)	0.128* (0.0670)
NumChoices=3	0	0
12-SecondGame		0.018 (0.0416)
GameCounter		0.002 (0.0017)
Model Fit (QICC)	103.104	106.727

(b) Games with dynamic prices

Table 4: GEE for the dependent variable *Realized Value*, studying the effect of *NumChoices* and controlling for *12-SecondGame* and *GameCounter*. Standard errors are given in parentheses under the coefficients. The individual coefficient is statistically significant at the *10%level, the **5% level, the ***1% level, and at the ****0.1% level.

The realized value increases a lot as we go from 3 to 4 to 5 choices, and then essentially plateaus as we go from 5 to 6 choices. One possible explanation for the effect we are seeing when going from 5 to 6 choices is that the disadvantage from adding more cognitive load outweighs or at least roughly equals the theoretical benefits of having one more choice available. For more insights, we now turn to the statistical data analysis.

Consider Tables 4 (a) and (b), where we present the results of the regression analysis. The coefficients for *NumChoices* are with respect to *NumChoices=3*. In Table 4 (a) column (1), we see that the coefficients for *NumChoices=4* and *NumChoices=5* are positive, and the coefficient for *NumChoices=5* is statistically significant at $p < 0.05$. Thus, increasing the number of choices has a statistically significant positive effect on the users' realized value. While the coefficient for *numChoices=6* is smaller than for *numChoices=5*, indicating a decrease in the realized value, we have verified that this decrease is not statistically significant ($p = 0.246$).

Now, consider Table 4 (b) column (1), where we present the same analysis for the games with dynamic prices. Here we see that the coefficients for *NumChoices=4*, *NumChoices=5*, and *NumChoices=6* are also all positive and all are statistically significant ($p < 0.1$ for 4 choices, and $p < 0.001$ for 5 and 6 choice). Furthermore, the coefficient for *NumChoices=6* is slightly larger than the coefficient for *NumChoices=5*, but this effect is not statistically significant ($p = 0.645$). Thus, for the game with fixed prices, increasing the number of choices definitely increases the user's realized value, but the effect plateaus when moving from 5 to 6 choices.

In column (2) of Table 4 (a) and (b) respectively, we add the two covariates *12-SecondGame* and *GameCounter* to the analysis to test the robustness of the results (adding them separately to the regression leads to the same qualitative results) The covariate *12-secondGame* indicates whether the user was playing a 7-second or a 12-second game. However, we see that it has no statistically significant effect on the realized value, which may be due to the fact that in each treatment, the users always started with four 12-second games before playing four 7-second games. Thus, by the time they played the 7-second games they already had more practice. We also add the covariate *GameCounter* to the analysis, which represents the number of games the user has already played up to that point in the overall experiment, and thus controls for possible learning effects. We see that this covariate has no statistically significant effect as well. Obviously, moving from 12-second games to 7-second games makes the game more difficult, while learning effects should make the game easier for the user over time. If either of those effects was present, our analysis suggests that they cancel each other out. Note that while adding those covariates, the results regarding *NumChoices* remain qualitatively unchanged. Thus, we obtain the following results regarding Hypotheses 1 and 2:

Result 1 (Number of Choices)**.** *We reject the null hypothesis in favor of H1, i.e., the realized value per game significantly increases as we increase the number of available choices from 3 to 4 to 5. Regarding H2, we cannot reject the null hypothesis. While for the games with fixed prices, the realized value per game seems to decrease as we go from 5 to 6 choices, this effect is not statistically significant. Furthermore, for games with dynamic prices, the realized value seems to plateau at 5 choices. While we observe a minimal increase in realized value when moving from 5 to 6 choices, this effect is not statistically significant.*

4.2. Fixed vs. Adaptive Choice Sets

We now move on to the analysis of the data from Experiment 2 where we studied the two design levers *Fixed vs. Adaptive Choice Sets*, and *UIOptimization*. For this experiment, we fixed the number of available choices to four and only considered dynamic prices. The design lever *Fixed vs. Adaptive Choice Sets* is based on the idea that we would like to present users with different choice sets in different situations. While an intelligent agent can never truly *know* a user's current value for high bandwidth, in some domains like the smartphone domain, we get a lot of signals from the user over time that can be used as input to a learning algorithm. Thus, we could learn a mapping from context to a value *estimate*. For example, when a user is currently listening to Internet radio then he may be more likely to choose a high bandwidth choice when presented with the bandwidth market UI, compared to situations when he is updating his Facebook status. Over time, the application could learn this behavior, inferring that the user has a higher value for

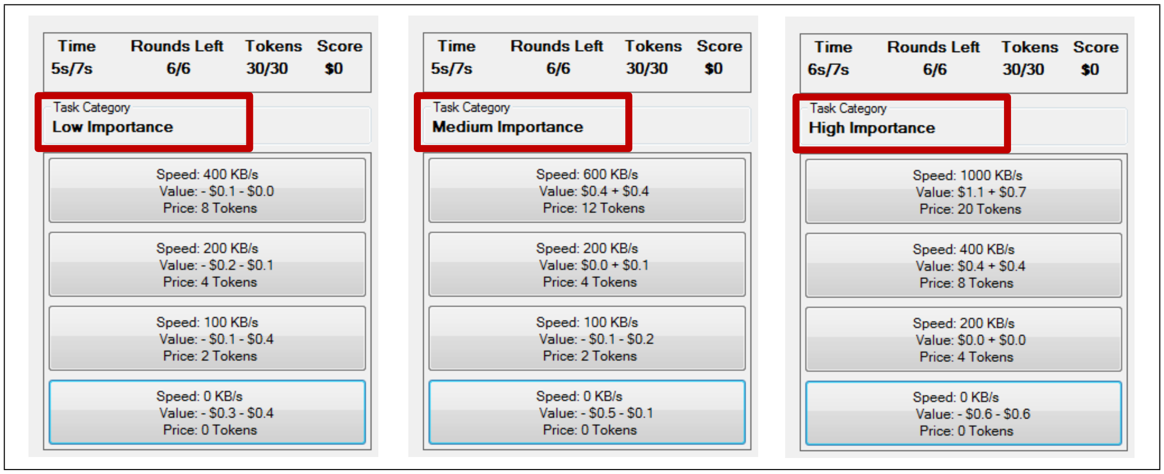

Figure 3: Adaptive Choice Sets: 3 different screenshots demonstrating the adaptive choice set idea. The users are offered a different set of choices (i.e., speed levels) depending on the current task category.

bandwidth when using the radio application. Thus, when presenting the user with the market UI in such a high-value situation, the application could then offer the user more choices at the higher end of the bandwidth spectrum and fewer choices at the lower end, enabling the user to make even more fine-grained decision.

The algorithm for finding the "optimal adaptive choice sets" works similarly as described before, except that now the algorithm takes into account that the choice set composition can be different for each category (i.e, the design space has grown cubically). Consider Figure 3 where we display three different screenshots, illustrating the three different choice sets offered to the user for the three different categories. We see that, as expected, the optimal choice sets include more low speed choices for low value categories, and more high speed choices for high value categories. Thus, on the one hand, the choices are now better tailored to the individual decision situation. On the other hand, the user now has to deal with the fact that the choices available to him (and thus also the prices) keep changing every round. The question is whether both effects taken together are positive or negative for the user's *RealizedValue*.

Consider Tables 5 (a) and (b), where we compare the optimal value of the game with the average value realized by the users. In this analysis, we separate the games without behavioral optimization from those with behavioral optimization. We see that in both cases, the optimal value of the game slightly increased when going from fixed to adaptive choice sets. This was expected, because the choice sets are now tailored to the three states *low importance*, *medium importance*, and *high importance*. However, while the optimal value of the game only increased by 0.02 in both cases, we see that the average realized value increased much more, by 0.12 and 0.04 in cases (a) and (b) respectively. Thus, presenting the users with adaptive choice sets had a positive effect on the realized value,

	Optimal Value	Realized Value
fixed choice sets	1.01	0.44
adaptive choice sets	1.03	0.56

(a) Without behavioral optimization

	Optimal Value	Realized Value
fixed choice sets	0.77	0.37
adaptive choice sets	0.79	0.41

(b) With behavioral optimization

Table 5: Optimal Values and Realized Values.

over and above the effect of just increasing the game's optimal value. To see if this effect is also statistically significant, we must must look at the difference between optimal value and realized value, and analyze whether the decrease in this difference is statistically significant.

Now consider Tables 6 (a) and (b), where we show the results of fitting the GEE to the data of Experiment 2, where the dependent variable is *OptimalValueMinusRealizedValue*. We see that the coefficient for *AdaptiveChoiceSets?* is negative (i.e., reducing the difference), but only statistically significant in case (a), i.e., for the games without behavioral optimization ($p < 0.1$). Thus, the data provides evidence that using adaptive choices indeed increased the users' average *RealizedValue*, over and above the positive effect on the optimal value. Apparently, the negative effect of having more variability was significantly smaller than the positive effect of being able to make better decision, as the available choices are better tailored to the specific situations. However, the effect was only statistically significant in case (a), and even there only with a relatively small statistical significance. Thus, we must be cautious when drawing conclusions based on this result.

Result 2 (Fixed vs. Adaptive Choice Sets)**.** *We reject the null hypothesis in favor of H3, i.e., the realized value is significantly higher with adaptive choice sets, compared to fixed choice sets. However, the effect is only statistically significant in the games without behavioral optimization, and even there, the significance is relatively low ($p = 0.098$).*

Factors/Covariates	(1)
Intercept	0.566**** (0.0578)
AdaptiveChoiceSets?	-0.090* (0.0376)
Model Fit (QICC)	65.534

(a) Without behavioral optimization

Factors/Covariates	(1)
Intercept	0.404**** (0.0398)
AdaptiveChoiceSets?	-0.024 (0.0442)
Model Fit (QICC)	45.215

(b) With behavioral optimization

Table 6: GEE for dependent variable *OptimalValueMinusRealizedValue*, studying the effect of *AdaptiveChoiceSets*. Standard errors are given in parentheses under the coefficients. The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0.1% level.

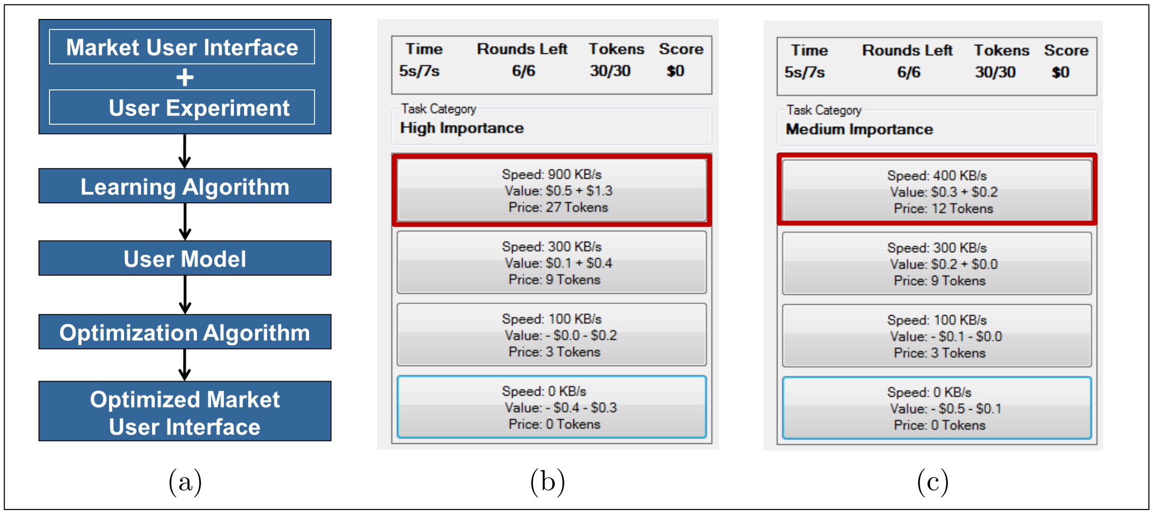

Figure 4: (a) Market UI Optimization Method. (b) A sample UI optimized assuming perfectly rational play. (c) A sample UI optimized assuming behavioral play.

4.3. UI Optimization for Rational vs. Behavioral Play

For the design lever *UIOptimization* we compare two different UIs, one optimized for perfectly rational play, and one optimized for behavioral play. For the behavioral optimization, we first built a behavioral model based on the data from Experiment 1. We computed different likelihood-maximizing λ-parameters for the quantal response model depending on 1) the total number of choices in the particular game, 2) the number of choices left in a particular round, and 3) whether prices were fixed or dynamic. Then we solved the resulting MDP, where now Q-values are computed assuming that the user will follow the "behavioral strategy" when playing the game. Finally, we selected the UI with the highest expected value according to this "behavioral MDP." Figure 4 (a) shows a diagram illustrating our market UI optimization methodology.

To get some intuition for what happens under behavioral optimization, consider Figures 4 (b) and (c) where we display two sample UIs, one optimized for perfectly rational play, and one optimized for behavioral play. Note that both UIs are the result of a computational search algorithm. The only difference between the two UIs is the top choice: the UI that was optimized for perfectly rational play gives the user the 900KB/s choice, while the UI that was optimized for behavioral play gives the user the 400KB/s choice. This result is understandable in light of how the UI optimization algorithm works and the behavioral vs. optimal user model. The quantal response model assigns each action a certain likelihood of being chosen, corresponding to the Q-values of those actions. Now, consider the top choice in Figure 4 (b), which has a high value, but which can also cost between 9 and 27 tokens (this is a game with dynamic prices). Thus, in the worst case, the user spends 27

Factors/Covariates	(1)
Intercept	0.444**** (0.0578)
BehavioralOptimization?	-0.075 (0.0545)
Model Fit (QICC)	63.060

(a) With fixed choice sets

Factors/Covariates	(1)
Intercept	0.558**** (0.0470)
BehavioralOptimization?	-0.148**** (0.0366)
Model Fit (QICC)	45.688

(b) With adaptive choice sets

Table 7: GEE for dependent variable *RealizedValue*, studying the effect of *BehavioralOptimization*. Standard errors are given in parentheses under the coefficients. The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0.1% level.

out of his 30 tokens with one click, and then has only 3 tokens left for the remaining 5 rounds. Even if it is very unlikely that the user selects this action, the negative effect of an occasional mistake would be very large. Consequently, the UI optimized for behavioral play shown in Figure 4 (c) does not have such high-value high-cost choices, reducing the negative effect of mistakes.

Now, consider Tables 7 (a) and (b) where we show the effect of the design lever *BehavioralOptimization?* on the dependent variable *RealizedValue*, separated into the two cases with fixed choice sets and with adaptive choice sets. First, we see that the coefficient for *BehavioralOptimization?* is negative in both cases, and highly statistically significant ($p < 0.001$) in case (b). Thus, the UI optimization assuming behavioral play did not have a *positive* but a *negative* effect on *RealizedValue*, and we obtain the following result:

Result 3 (UI Optimization)**.** *We cannot reject the null hypothesis in favor of H4. Instead, we find that using the behaviorally optimized UI leads to a realized value which is not higher but lower compared to using the UI optimized for rational play. This effect not statistically significant for the games with fixed choice sets, but it is highly statistically significant for the games with adaptive choice sets.*

This result is very surprising, in particular that the behavioral UI optimization had a *negative* effect on *RealizedValue*. Upon finding this result, we hypothesized that the quantal response model was too simple for a UI optimization in our domain, ignoring some important behavioral factors. Given prior behavioral research, possible candidate factors were *loss aversion* and *position effects*. The goal of the analysis in the next section is to find empirical support for our hypothesis that behavioral factors which we omitted in our UI optimization had a significant impact on users' decisions.

5. Behavioral Decision Analysis

In this section, we analyze users' individual actions in each *round* of a game to understand which behavioral factors influence users' decision making. To simplify the analysis of this round-based data, we study the binary dependent variable *OptChoice*, which is 1 if the user clicked on the *optimal* choice, and 0 otherwise. The analysis in this Section is only based on data from Experiment 1 (see Appendix Section B for details on the data set). Note that there is no widely accepted definition of standardized coefficient estimates for logistic regression models. Thus, when reporting regression results using the logit link function, we only report the non-standardized coefficient estimates B and the corresponding odds ratios $Exp(B)$.

5.1. User-specific Factors: Degree of Rationality

We first test whether individual users exhibit significant differences in their play according to the quantal response model. We compute a separate maximum-likelihood parameter λ_i for each user i. This parameter can be seen as measuring how "rational" a user's play was. In fact, the users exhibited large differences, with a minimum λ of 3.9, a maximum of 9.0, and a median of 6.8. Table 8 presents the regression results for *OptChoice*. In column (1), we see that the parameter *Lambda* has a statistically significant effect ($p < 0.001$). Looking at the odds ratio (Exp(B)), we see that the odds of choosing the optimal choice are 16% higher for a user with $\lambda = x$ compared to a user with $\lambda = x - 1$. Thus, for the analysis of *OptChoice* it is important to control for λ.

We also analyzed two other user-specific factors: *Age* and *Gender*. There was no statistically significant effect of *Age* on either *OptChoice* or *RealizedValue*. For *Gender*, there was no effect with respect to *RealizedValue*, but there was a small statistically significant effect ($p < 0.1$) on *Optchoice*: female users were slightly more likely to miss the optimal choice, but male users made bigger mistakes when they missed the optimal choice. However, the factor *Lambda* already captures user-specific cognitive differences, and thus we do not need to also control for *Gender* in the regression analyses.

5.2. Q-Value Differences

We now analyze the factor *QvalueDiff* which denotes the difference between the Q-values of the best and second-best action. In column (2) in Table 8 we see that *QvalueDiff* is statistically significant ($p < 0.001$) with an odds ratio of 354. Note that this is the odds ratio for a *one unit change* in the Q-value difference. Yet, in our data, the mean of the Q-value difference is 0.11. The odds ratio for a change of 0.1 is 1.8. Thus, holding *Lambda* constant, if the Q-value difference between the best and second-best choice increases by 0.1, the odds for choosing the optimal choice increase by 80%.

Factors	(1)		(2)	
	B	Exp(B)	B	Exp(B)
Intercept	-0.816**** (0.1408)	0.442****	-1.529**** (0.1593)	0.217****
Lambda	0.150**** (0.0180)	1.162****	0.161**** (0.0197)	1.175****
QvalueDiff			5.868**** (0.4353)	353.713****
Fit (QICC)	(3771.953)		(3589.063)	

Table 8: GEE for dependent variable *OptChoice*, studying *Lambda* and *QvalueDiff*. Standard errors are given in parentheses under the coefficients. The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0.1% level.

5.3. UI Design: Number of Choices

We now study how the UI design affects users' ability to make optimal choices. Consider Table 9 column (1), where we add *NumChoices* to the regression. This factor denotes the number of choices in the game (i.e., 3, 4, 5, or 6 choices). We see that the factor has a large negative effect on *OptChoice*, and the effect is highly statistically significant ($p < 0.001$). Holding all other factors constant, increasing the number of choices by 1 reduces the odds for selecting the optimal choice by 32%. Naturally, a more complex UI (i.e., more choices) makes it harder for users to find the optimal choice. We also analyzed *NumChoicesLeft* which denotes the number of choices that were still affordable during a game situation, given the prices of the current choices and the user's budget. However, when controlling for *NumChoices* we found that *NumChoicesLeft* does not have a statistically significant effect on *OptChoice*.

5.4. Incomplete Search: Position Effects

By design, the game exhibits a strong ordering effect: the values of the choices decrease monotonically from top to bottom, as do the prices. Thus, it is conceivable that users scan the choices in a linear way, either from top to bottom or from bottom to top. Given that they are under time pressure, incomplete search effects may be expected, and prior research has shown that this can lead to significant position effects (Dumais, Buscher and Cutrell, 2010; Buscher, Dumais and Cutrell, 2010). We can control for positional effects by adding information about the position of the optimal choice to the regression. Consider column (2) in Table 9 where we added six indicator variables to the regression. *OptRelativeRank* denotes the "relative rank" or "relative position" of the optimal choice, taking into account

Factors/Covariates	(1)		(2)		(3)	
	B	Exp(B)	B	Exp(B)	B	Exp(B)
Intercept	0.283 (0.2182)	1.327	-0.495* (0.2489)	0.610*	-0.616*** (0.2339)	0.540***
Lambda	0.167**** (0.0206)	1.181****	0.162**** (0.0223)	1.176****	0.158**** (0.0238)	1.171****
QvalueDiff	5.062**** (0.4297)	157.888****	4.421**** (0.5061)	83.196****	4.595**** (0.4989)	98.962****
NumChoices	-0.391**** (0.0446)	0.677****	-0.087* (0.0487)	0.916*	-0.065 (0.0583)	0.937
OptRelativeRank=5			-3.853**** (0.9808)	0.021****	-4.046**** (1.0396)	0.017****
OptRelativeRank=4			-1.893**** (0.4499)	0.151****	-1.853**** (0.4959)	0.157****
OptRelativeRank=3			-1.201**** (0.2706)	0.301****	-1.188**** (0.3382)	0.305****
OptRelativeRank=2			-0.614** (0.2807)	0.541**	-0.522 (0.3351)	0.593
OptRelativeRank=1			-0.160 (0.2283)	0.852	-0.170 (0.2512)	0.844
OptRelativeRank=0			0	1	0	1
OptimalChoiceNegative=1					-1.299****	0.273**** (0.2247)
CurrentCategory=2					1.532**** (0.2059)	4.626****
CurrentCategory=1					0.033 (0.1295)	1.034
CurrentCategory=0					0	1
Goodness of Fit (QICC)	3476.044		3345.116		3288.243	

Table 9: GEE for dependent variable *OptChoice*, studying UI complexity, position effects, and loss aversion. Standard errors are given in parentheses under the coefficients. The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0.1% level.

the currently unavailable choices (and counting from the top starting at 0). Consider a game with 6 choices as an example. If two choices are unavailable such that there are still 4 choices left, and the optimal choice is the third from the top, then the absolute rank of that choice is 2, but the relative rank is 0. We use the relative rank rather than the absolute rank for two reasons. First, using the absolute rank would not allow us to consider games with different number of choices in one regression. Second, as more and more choices become unavailable during a game (as the user depletes his budget), the relative rank keeps adjusting, to reflect that a user doesn't need to scan the non-available choices, while the absolute rank doesn't adjust. Thus, going forward, we use the relative rank in the regressions analyses. However, we have also performed the same analyses using absolute rank and obtained qualitatively similar results.

In column (2) of Table 9 we see that *OptRelativeRank* has a very strong, and highly statistically significant negative effect on *OptChoice*. Note that all coefficient estimates are relative to *OptRelativeRank*=0. The lower the rank of the optimal choice, the less likely the users were to choose the optimal action. As we go from *OptRelativeRank*=0 to *OptRelativeRank*=5, the coefficients decrease monotonically, and except for *OptRelativeRank*=1, all of the effects are statistically significant. Compared to the case when the optimal choice has rank 0, holding everything else constant, if *OptRelativeRank*=4 the odds of choosing the optimal action decrease by 84%, and if *OptRelativeRank*=5, the odds decrease by 98%. Thus, in particular for the very low ranks, the position effect is indeed very strong and highly statistically significant ($p < 0.001$), and because our user model did not take it into account, this presents a possible explanation for why the UI optimization failed.

5.5. Loss Aversion

Loss-aversion, i.e., people's tendency to avoid losses more than they appreciate same-sized gains, is a well-known effect in behavioral economics (Tversky and Kahneman, 1991). Thus we hypothesized to find it in our data as well. Consider column (3) of Table 9 where we added *OptimalChoiceNegative* to the regression, an indicator variable that is 1 when the value of the optimal choice is negative, and 0 otherwise. Additionally, we also added the factor *CurrentCategory* to the regression, controlling for the different value distributions in different game situations. *OptimalChoiceNegative* has a large negative coefficient, and is statistically significant ($p < 0.001$). Thus, whether the optimal choice has a positive or negative value has a large effect on users' behavior, providing strong evidence for the loss aversion hypothesis. In a more detailed statistical analysis of this effect (regression table omitted), we also found that users are particularly likely to make a mistake when the optimal choice has a negative value and the choice right above it has a positive value. We consider this to be the most convincing evidence of users' loss aversion, as this shows that a large driver of their decision is whether the absolute value of a choice is positive or negative. Note that this last effect cannot be attributed to a position effect because *OptRelativeRank* is part of the regression and we are thus already controlling for the position effect.

An alternative explanation for the observed effect could be classical risk aversion, which is based on diminishing marginal utility of wealth (Koeszegi and Rabin, 2007). However, in the 40-50 games that users played per experiment, they repeatedly faced many small-scale risks with almost no effect on their overall wealth. Thus, loss aversion is a more a convincing explanation for the observed behavior than risk aversion.

Another possible explanation for the observed effect could be myopia. It is conceivable that users have a limited look-ahead horizon when making decisions during the game, and thus do not fully account for the effect of running out of budget towards the end of the game. However, in further statistical analyses we could not find evidence for this hypothesis. In particular, we added the factors *CurrentTimeStep* and *CurrentBudget* to the regression, but found no statistically significant effect for either factor.

6. Towards Personalized Market User Interfaces

We have seen that behavioral factors such as position effects and loss aversion play a significant role in users' decision making, offering potential answers to the question why the behavioral UI optimization failed. We now come back to the design lever *UI Optimization* that we studied in Experiment 2. In further analyses of *OptChoice* (see Table 13 in the Appendix Section C), we find that the behavioral UI optimization indeed made the decision problem easier for the users: they were 17% more likely to select the optimal choice when using the UI optimized for behavioral play. Given that the users made better choices but their *RealizedValue* still decreased, this suggests that the UI optimization eliminated too many valuable choices. In some sense, it was "too aggressive."

Behavioral Optimization?	*Optimal Value*	*Realized Value*
no	1.02	0.50
yes	0.78	0.39

Table 10: UI optimization: Effects on optimal and realized value.

Now consider Table 10, which shows what happened to *OptimalValue* and *RealizedValue* under the behavioral optimization (the values are averaged between the games with fixed and adaptive choice sets). By using the behavioral optimization, we decreased the optimal value (achievable for a perfectly rational player) from \$1.02 to \$0.78. Thus, we "took away" approximately \$0.24 per game. Note that we never expected the users to come even close to the optimal values, but instead we expected them to do better using the behaviorally optimized UI such that the *Realized Value* would actually increase. However, as we can see in the last column of Table 10, the *Realized Value* also dropped from \$0.50 to \$0.39. Relative to the optimal value, the users did better in the re-optimized game – but in absolute terms they did worse.

Factors/Covariates	(1)
Intercept	0.360**** (0.0331)
Lambda	0.048**** (0.0068)
AdaptiveChoices=1	0.061 (0.0424)
BehavioralOptimization=1	-0.172**** (0.0371)
Model Fit (QICC)	30.050

Table 11: GEE for dependent variable *RealizedValue* for ***SmallLambda*=0**, studying the effect of *BehavioralOptimization*.

Factors/Covariates	(1)
Intercept	-0.308**** (0.0377)
Lambda	0.186**** (0.0094)
AdaptiveChoices=1	0.099* (0.0572)
BehavioralOptimization=1	-0.068 (0.0495)
Model Fit (QICC)	74.808

Table 12: GEE for dependent variable *RealizedValue* for ***SmallLambda*=1** studying the effect of *BehavioralOptimization*.

A potential explanation is that by coincidence, the users in Experiment 2 acted "more rationally" than the users in Experiment 1. However, the best fitting λ-parameters for experiments 1 and 2 are very similar, and thus, the data does not support this hypothesis. Yet, we found another unexpected result regarding users' level of rationality in Experiment 2. As before, we computed a λ_i-parameter for each user, as well as one λ corresponding to the best fit across all users. Next, we computed a binary variable $SmallLambda_i$ for each user i which is 1 if a user's λ_i is smaller than the *average* λ. Thus, *SmallLambda* denotes whether a user belongs to the *more rational* or to the *less rational* group of users.

Now consider Tables 11 and 12, where we study the effect of *BehavioralOptimization* on *RealizedValue*, separating users into the *more rational users* (on the left) and the *less rational users* (on the right). For *SmallLambda*=0 (the more rational users), the effect of *BehavioralOptimization* is particularly negative: for those users we made the game a lot worse by doing the re-optimization. However, for *SmallLambda*=1 (the less rational users), the effect of *BehavioralOptimization* is close to zero, and not statistically significant. Thus, for less rational users, the behaviorally-optimized UI was easier to use, but the resulting *RealizedValue* was practically the same.

This finding suggests a new research direction on *personalized market user interfaces*, with the goal to tailor the UI to the capabilities, needs, and preferences of individual users. To achieve this, we must access or observe user-specific, behavioral and non-behavioral data. This is available in many domains, in particular in the smartphone domain. Once we have an estimate of a user's "degree of rationality," we can provide each user with a market UI that is specifically optimized for that particular user.

7. Conclusion

In this paper, we have introduced a new research agenda on *market user interface design*. Our long-term goal is to understand how UI design choices for markets affect users' abilities to make good economic decisions, and how we can develop automated methods to optimize market UIs. In studying this question, it is crucial to take a behavioral approach, deviating from a perfectly rational agent model. Thus, our research explores a design space in which human limited cognition meets computing.

We ran a behavioral economics lab experiment, testing the effect of different market UI design levers. In regard to the number of choices, we found that the realized value increases as we go from 3 to 4 to 5 choices, with no significant effect going from 5 to 6 choices. In future experiments, we will also test 7 and 8 choices, to see if the realized value increases further or ultimately decreases again. For the design lever *Fixed vs. Adaptive Choice Sets*, we found that the realized value is significantly higher with adaptive choice sets.

Finally, the most interesting design lever was the behavioral UI optimization. An unexpected result is that the realized value was *lower* when using the behavioral UI optimization. This suggests that our user model, based on the quantal response model, was too simplistic to accurately predict user behavior. In future research, we will consider other behavioral models that are better supported by neuroeconomic experiments like the drift-diffusion model by Fehr and Rangel (2011).

In a subsequent decision analysis, we found that our model ignored important behavioral factors like loss aversion and position effects. Yet, the most intriguing result concerns how *less rational* and *more rational* users differed regarding the effect of the UI optimization. While there was no significant difference regarding the realized value for the less rational users, the more rational users lost a lot of value in the UI optimization due to precluded opportunities. This result points towards the need to estimate each individual user's level of rationality based on behavioral data obtained over time, to generate personalized market UIs. Taking this idea a step further, we could also take a user's personal value for time into account, when generating a personalized UI for this user, or even when automatically making decisions on a user's behalf. Certainly, learning users' preferences, or an individual level and on an aggregate level, will be important to automatically generate market UIs that maximize users' welfare. Thus, there are still many opportunities for research at the intersection of market design, intelligent agents, UI design, and behavioral economics, ranging from better behavioral models, to algorithms for learning user preferences and automated UI optimization.

References

Buscher, Georg, Susan Dumais, and Edward Cutrell. 2010. "The Good, the Bad, and the Random: An Eye-Tracking Study of Ad Quality in Web Search." In *Proceedings of the 33rd Annual International ACM SIGIR Conference*. Geneva, Switzerland.

Chabris, Christopher F., David I. Laibson, Carrie L. Morris, Jonathon P. Schuldt, and Dmitry Taubinsky. 2009. "The Allocation of Time in Decision-Making." *Journal of the European Economic Association*, 7: 628–637.

Chen, Li, and Pearl Pu. 2010. "Experiments on the Preference-based Organization Interface in Recommender Systems." *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(1): 1–33.

Dumais, Susan, Georg Buscher, and Edward Cutrell. 2010. "Individual Differences in Gaze Patterns for Web Search." In *Proceedings of the Information Interaction in Conext Symposium (IIiX)*. New Brunswick, NJ.

Fehr, Ernst, and Antonio Rangel. 2011. "Neuroeconomic Foundations of Economic Choice–Recent Advances." *Journal of Economic Perspectives*, 25(4): 3–30.

Gabaix, Xavier, David Laibson, Guillermo Moloche, and Stephen Weinberg. 2006. "Costly Information Acquisition: Experimental Analysis of a Boundedly Rational Model." *American Economic Review*, 96 (4): 1043–1068.

Gajos, Krzysztof Z., Daniel S. Weld, and Jacob O. Wobbrock. 2010. "Automatically Generating Personalized User Interfaces with Supple." *Artificial Intelligence*, 174: 910–950.

Hauser, John R., Glen L. Urban, Guilherme Liberali, and Michael Braun. 2009. "Website Morphing." *Marketing Science*, 28(2): 202–223.

Horvitz, Eric, and Matthew Barry. 1995. "Display of Information for Time-Criticial Decision Making." In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI)*. Montreal, Canada.

Iyengar, Sheena S., and Mark R. Lepper. 2000. "When Choice is Demotivating: Can One Desire Too Much of a Good Thing?" *Journal of Personality and Social Psychology*, 79(6): 995–1006.

Johnson, Eric J., John W. Payne, and James R. Bettman. 1988. "Information Displays and Preference Reversals." *Organizational Behavior and Human Decision Processes*, 42(1): 1–21.

Knijnenburg, Bart P., Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. "Explaining the User Experience of Recommender Systems." *User Modeling and User-Adapted Interaction (UMUAI)*, 22: 1–64.

Koeszegi, Botond, and Matthew Rabin. 2007. "Reference-Dependent Risk Attitudes." *American Economic Association*, 97(4): 1047–1073.

Malhotra, N. K. 1982. "Information Load and Consumer Decision Making." *Journal of Consumer Research: An Interdisciplinary Quaterly*, 8(4): 419–430.

McKelvey, Richard, and Thomas Palfrey. 1995. "Quantal Response Equilibria for Normal Form Games." *Games and Economic Behavior*, 10: 6–38.

Miller, George A. 1956. "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information." *The Psychological Review*, 63: 81–97.

Nelder, John, and Robert Wedderburn. 1972. "Generalized Linear Models." *Journal of the Royal Statistical Society*, 135 (3): 370–384.

Puterman, Martin L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* New York, NY:John Wiley & Sons, Inc.

Rysavy Research. 2010. "Mobile Broadband Capacity Constraints and the Need for Optimization." http://www.rysavy.com/papers.html.

Schwartz, Barry. 2005. "Can There Ever be too Many Flowers Blooming?" *Culture Choice*, 3: 1–26.

Seuken, Sven, David C. Parkes, and Kamal Jain. 2010. "Hidden Market Design." In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI).* Atlanta, GA.

Seuken, Sven, David C. Parkes, Eric Horvitz, Kamal Jain, Mary Czerwinski, and Desney Tan. 2011. "Market User Interface Design [Extended Abstract]." In *Proceedings of the Second Conference on Auctions, Market Mechanisms and Their Applications.* New York, NY.

Seuken, Sven, David C. Parkes, Eric Horvitz, Kamal Jain, Mary Czerwinski, and Desney Tan. 2012. "Market User Interface Design." In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC).* Valencia, Spain.

Seuken, Sven, Denis Charles, Max Chickering, and Sidd Puri. 2010*a*. "Market Design and Analysis for a P2P Backup System." In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC).* Cambridge, MA.

Seuken, Sven, Kamal Jain, Desney Tan, and Mary Czerwinski. 2010*b*. "Hidden Markets: UI Design for a P2P Backup Application." In *Proceedings of the Conference on Human Factors in Computing Systems (CHI).* Atlanta, GA.

Simon, Herbert A. 1971. "Designing Organizations for an Information-Rich World." In *Computers, Communication, and the Public Interest.* Ed. Martin Greenberger. Johns Hopkins Press.

Teschner, Florian, and Christof Weinhardt. 2011. "Evaluating Hidden Market Design." In *Proceedings of the 2nd Conference on Auctions, Markets, and Market Applications (AMMA)*. New York, NY.

Thaler, Richard H., Cass R. Sunstein, and John P. Balz. 2010. "Choice Architecture." http://ssrn.com/abstract=1583509.

Tversky, Amos, and Daniel Kahneman. 1991. "Loss Aversion in Riskless Choice: A Reference Dependent Model." *Quarterly Journal of Economics*, 106(4): 1039–1061.

Wright, James R., and Kevin Leyton-Brown. 2010. "Beyond Equilibrium: Predicting Human Behavior in Normal-Form Games." In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI)*. Atlanta, GA.

Appendix

A. Computation of the Realized Value (Users' Game Scores)

To analyze the effect of the four design levers on users' performance, we could simply measure and compare the average game scores that users achieved per treatment. However, that measure is very noisy due to the high degree of randomness in the game itself. To account for this, we compute a different measure, removing the randomness inherent to the game and the user's luck as much as possible. The main idea is to take the expected optimal value of the game and subtract the expected value loss due to the user's actions. Our formula to compute the *RealizedValue* is as follows:

$$RealizedValue = ExpectedOptimalValue - \frac{ExpectedOptimalValue + 7.4}{OptimalScoreThisGame + 7.4} \cdot UserValueLoss$$

The *ExpectedOptimalValue* is the expected a priori value for playing the game optimally, without knowing the realization of the state uncertainties. Thus, this is simply the value of the corresponding MDP. The *OptimalScoreThisGame* is the score a player could have achieved in this particular instantiation of the game, had he followed the optimal policy (not knowing the future). Finally, the *UserValueLoss* is the sum of the differences between the Q-value of the optimal choice in each round and the Q-value of the choice selected by the user. Thus, *ExpectedValueLoss* is a measure of how much value a user playing this particular strategy would lose in this game instance on average (thus, also removing the randomness due to cases where the user just got lucky).

We add 7.4 to *ExpectedOptimalValue* and to *OptimalScoreThisGame* to normalize both values such that they cannot be negative. Then we scale each game's *UserValueLoss* by the ratio of these two numbers such that the same kind of mistake of a user leads to the same penalty, independent of whether the user got lucky with this game instance or not (high or low *OptimalScoreThisGame*). This gives us a normalized measure for value loss. Then we subtract this normalized measure from the *ExpectedOptimalValue* and obtain the *RealizedValue*. Note that, if we let the number of games played go to infinity, then the average game scores would approach *Realized Value*. However, with just a few hundred games played per design lever, the impact of the game's randomness and the user's luck on the game scores can be large, which is why we use *RealizedValue* instead.

B. Data Selection for Behavioral Decision Analysis

From Experiments 1 and 2, we obtained 10,176 data points for the exogenous time treatments (7-second and 12-second games). Because we tested four different design levers, there is a lot of variance in the data. To most cleanly identify the behavioral decision factors, we only study the data resulting from the analysis of the design lever *Number of Choices*. Thus, for the analysis presented in Section 5, we only consider the data points from Experiment 1, and only for games with fixed prices and with a 7-second or a 12-second time limit, which leaves us with 3,456 data points. We exclude all cases with *timeStep*=6 because in the last round of a game, the optimal choice is always the highest-ranked choice still available, and thus the decision problem is trivial. This leaves us with 2,880 data points. Furthermore, we exclude 7 cases where only one choice was left (nothing to decide), and 10 cases where only two choices was left (very unusual decision situation, often because the user ran out of budget due to a mistake). This leaves us with 2,863 data points. Moreover, a numerical rounding error in the software lead to a few cases where the values on the available choices were in the wrong order. Excluding those cases leaves us with 2,786 data points. Lastly, we exclude another 30 cases where a user let the timer run out (and thus the bottom-choice was automatically selected), which leaves us with a total of 2,756 cases (i.e., round). Note that we analyze games with a 7-second and with a 12-second time limit in one analysis, because we could not find a statically significant effect of the time limit on users' decision performance.

C. The Effect of Behavioral Optimization on OptChoice

Factors/Covariates	(1)	
	B	Exp(B)
Intercept	-0.080 (0.3607)	0.923
Lambda	0.193**** (0.0286)	1.213****
QvalueDiff	3.324**** (0.4165)	27.780****
NumChoicesLeft=4	-0.665 (0.0.1338)	0.514
NumChoicesLeft=3	0	1
OptRelativeRank=3	-3.295**** (0.4882)	0.037****
OptRelativeRank=2	-1.010**** (0.2760)	0.364****
OptRelativeRank=1	-0.233 (0.2252)	0.792
OptRelativeRank=0	0	1
AdaptiveChoices=1	0.096 (0.0808)	1.100
BehavioralOptimization=1	0.158** (0.0719)	1.171**
Model Fit (QICC)	3329.030	

Table 13: GEE for dependent variable *OptChoice* studying the effect of *BehavioralOptimization*. Standard errors are given in parentheses under the coefficients. The individual coefficient is statistically significant at the *10% level, the **5% level, the ***1% level, and at the ****0.1% level.