Market User Interface Design*

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

September 8, 2011

PRELIMINARY DRAFT! COMMENTS WELCOME!

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 levers on user behavior and market performance, along with considerations of behavioral factors including loss aversion, incomplete search, and position effects. Finally, we fit a quantal-best response model to users' actions and evaluate an optimized market user interface.

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

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 can be quite a challenge given limited user attention and exacerbated by markets that can easily present users with a large number of choices. Indeed, recent research has shown that having more choices does not always lead to better outcomes. For example, Iyengar and Jiang (2003) show that more choices in employees' 401(k) plans can lead to fewer participation and thus significant losses in savings. Choi et al. (2004)

*Part of this work was done while the first author was an intern at Microsoft Research. We are particularly thankful to Alvin E. Roth and Yiling Chen for feedback on this work. We also thank seminar participants from the Harvard EconCS group, and from AMMA'11 for very useful feedback. Seuken gratefully acknowledges the support of a Microsoft Research PhD Fellowship.

†Department of Informatics, University of Zurich, Binzmühlestraße 14, 8050 Zürich, Switzerland, 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, kamalj, marycz, desney}@microsoft.com

show that in general, 401(k) plan design has a huge impact on employer savings behavior. There may be many different reasons for this behavior, including cognitive overload resulting from more choices, emotional processes, or various decision-making biases. However, in this paper we are not primarily concerned about the cause of behavior but focus on modeling the effect of different user interfaces on users' decision-making performance.

Traditional economic models assume all agents to be perfectly rational, with unlimited time to make a decision and unbounded computational resources for deliberation. In reality, however, humans have cognitive costs, bounded time for decision making (because of opportunity costs) and bounded computational resources. We explicitly take these behavioral considerations into account, with the goal to design market user interfaces that make the decision-making task easier for the users and lead to better outcomes. Our approach is very much in line with the "choice architecture" idea put forwarded by Thaler et al. (2010). In their language, we are designing "choice architectures for electronic markets."

So far, the market design literature has largely ignored the intersection of market design and user interface design. However, we argue that this intersection is particularly important for at least four reasons. First, the UI is the first point of contact for a user interacting with a new market. Second, the choice of the UI constrains the design space for the market designer. Third, the UI defines how, and how well, users can express their preferences. And fourth, the complexity of the UI defines the cognitive load imposed on the user while interacting with the market. Thus, when designing an electronic market populated by end users, it is important to design the market and the user interface in concert, to jointly optimize along both dimensions.

1.1 Overview of Results

We propose a new research agenda on "market user interfaces" and present a principled study of the design space. A market UI can best be defined via two questions: first, what information is displayed to the user? Second, what choices/how many choices are offered to the user? The research question we want to answer is: *what is the optimal market user interface given that users have cognitive costs?* In evaluating the effectiveness of a market user interface we consider the ability of the market to efficiently allocate resources given user behavior.

We focus on the challenges in market user interface design for allocating 3G bandwidth, the demand for which is projected to continue to grow exponentially over the next few years. In particular, we present the results of a systematic, empirical exploration of the effect that different UI design levers have on users' performance in economic decision making. The experimental set-up considers a user with uncertainty about future value for resource allocation, an inter-temporal budget constraint, and a user interface that offers some number of choices of bandwidth in any given period, each for a particular price. Formally, the decision problem facing a user is modeled as a Markov Decision Process (MDP), the solution to which provides the gold standard against which we compare user behavior.

We first explore parts of the design space manually, by experimenting with varying the number of choices offered to users, and considering the effect of offering fixed vs dynamically changing prices. These results offer general insight, we think for the first time in such detail, into how well humans can determine optimal policies in MDPs under time pressure. Our findings indicate that users are surprisingly good at coming up with good decision polices for the sequential optimization problem. We show that their actions exhibit a high degree of rationality in the sense of being highly correlated with the Q-values of the game. However, we also show how various behavioral factors influence the users' decision making process. Some effects are particularly strong, including loss aversion which raises concerns about users general tendency, at least in some situations, to take short-term winnings

ignoring potential long-term losses.

In a second step, we then use computation to *automate* the market UI optimization process. Based on the results from the first experiment, we train a behavioral user model. In particular, we adopt a maximum-likelihood fit to a quantal best-response user model (Wright and Leyton-Brown (2010)), which is a well-studied model of behavioral decision making. The model is a single-parameter, softmax model, allowing for a range of behavior from random to best-response, where the true utility for each choice is induced as the solution to the MDP model of the user problem. Based on this maximum likelihood fit, we then feed this user model into an optimization algorithm, which is used to identify the optimal market UI given the learned behavioral model. A second experiment evaluates the effect of the re-optimization algorithm. Here we find that the re-optimization increased the user's probability of selecting the optimal choice. However, the data suggests that the re-optimization algorithm took away too much value, in particular for the *more rational users*, while no statistically significant effect was observed for the *less rational users*.

1.2 Related work

Prior research has identified a series of behavioral effects in users' decision making. Buscher et al. (2010) show that the amount of *visual attention* users spend on different parts of a web page significantly depends on the task type and the quality of the information provided. Dumais et al. (2010) show that these "gaze patterns" differ significantly from user to user, suggesting that different user interfaces 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. Thus, time is generally costly, and consequently 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) who show that customers are happier with the choices they make when offered 6 different flavors of jam compared to 24 different flavors of jam. 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*. Sarver (2008) derives a formal model of regret anticipation for situations where agents select an alternative from a menu of choices. While we are aware that many emotional factors play a role in decision making, in this paper we do not aim to study these directly. Instead, we aim to develop a methodology to model user behavior and optimize UIs given this observed user behavior.

Horvitz and Barry (1995) present a methodology for the optimal design of human-computer interfaces for time-critical applications in non-market-based domains. They introduce the concept of *expected value of revealed information*, trading-off the costs of cognitive burden with the benefits of added information. Johnson et al. (1988) show that the way information is displayed, in particular probability values (fractional vs. decimal), has an impact on user decision making and their information processing strategies. The authors briefly discuss the implications of their findings for the design of information displays.

In our own previous work (Seuken et al. (2010b)), we have introduced the goal of designing *simple* and *easy-to-use* interfaces for electronic markets, in particular for domains where users repeatedly make decisions of small individual value. In a different paper (Seuken et al. (2010a)), we present one detailed case-study of a novel market user interface for a P2P backup market. We demonstrate that it is possible to hide many of the market's complexities, while maintaining a market's efficiency. However, there we did not study the effect of *changing* aspects of a UI on a user's decision-making performance, which is the focus of this paper.

The work most closely related to ours is SUPPLE, introduced by Gajos et al. (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 a computational optimization problem, and find that automatically-generated UIs can lead to significantly better performance compared to manufacturer's defaults. While their approach is very much in line with our long-term goal of "automatic UI optimization", they optimize their interfaces for accuracy, speed of use, and user's subjective preferences for UI layouts. In contrast, we optimize for *decision quality* in market-based environments where users are dealing with values, prices, and budgets. We build a parameterized *behavioral* model of users while they build a model of users' pointing and dragging performance. A significant part of this paper is about determining which behavioral factors are most important for the effectiveness of decision-making in a market-based environment.

1.3 Outline

The remainder of the paper is structured as follows. In the next section we describe the design of the market game that is the basis of our experiment. After motivating the domain of bandwidth allocation for smartphones, we describe the implementation of the market game in detail. We describe how the game can be modeled as a Markov Decision Process, and how the quantal-response model can be used as a behavioral model that predicts user play in this domain. In Section 3 we describe the experiment design. This includes a discussion of the four different design levers, the time limits we imposed, the selection of the subject pool and the experimental set-up, as well as a detailed description of the different treatment variations across users. In Section 4 we present the results of our statistical data analysis. We first present the results based on analyzing users' decisions in individual rounds, which allows us to study which factors are most predictive for whether users find the optimal choice or not. Then we move on to the analysis of whole games, studying the effect of the four different design levers on users' Realized Efficiency. We conclude in Section 5.

2 Game Design: Bandwidth Allocation over Time

2.1 Setting: A 3G Bandwidth Market

We situate the experiment in the smartphone domain to give our participants some context for the game they are playing. Consider the 3G bandwidth needed to access the Internet on a smartphone. According to Rysavy Research (2010), the demand for 3G bandwidth will continue to grow exponentially over the next few years and that it will be infeasible for the network operators to update their infrastructure fast enough to satisfy future demands . Another sign that 3G bandwidth is getting scarce is that network providers like AT&T are beginning to drop their unlimited data plans. 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 data usage constraints via fixed upper limits (e.g., 200MB per month for one of AT&T's current data plans). 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. The current approach simply ignores this.

Imagine a market-based solution to the 3G 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., update their Facebook status). If we assume that users are willing to accept low performance now for high performance later, then we can optimize the allocation of bandwidth use by shifting excess demand to times of excess supply. Of course it is not possible that every user

Figure 1: Mockup of the Bandwidth Market UI.

gets high-speed Internet access all of the time. Instead, the users' choices must be limited somehow. One possibility to achieve this is by giving each user a fixed amount of virtual currency, assuming users pay a fixed $-amount for their data plans.

Consider Figure 1 which shows a mock-up application for a 3G bandwidth market. Let's assume 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. However, when there is excess demand and the user wants to access the Internet, then the screen as shown in Figure 1 pops up, requiring the user to make a choice. Each speed level has a different price (in tokens). For now, we assume that when a user runs out of tokens, he gets the lowest possible service quality (which could mean no access or some very slow connection). Note that we do not concern ourselves with the economics of this market, nor with the question as to whether users should be allowed to pay money to buy more tokens or not. Our goal is not to put forward this particular market design as the best solution for this domain. Instead, we merely use this hypothetical market application as a motivating domain for our experimental study.

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. In our lab experiment we studied the effect of changing various design parameters on how well users were able to make good decisions.

2.2 Game Design

Figure 2 shows a screenshot of the market game that we designed, mirroring the mockup of the market application from Figure 1. Each game has exactly 6 rounds. At the beginning of a game, a user always has 30 tokens available to spend over the course of the 6 rounds. In each round, the user has to select one of the choices. Each choice (i.e., a button in Figure 2) 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 $. The value represents the $

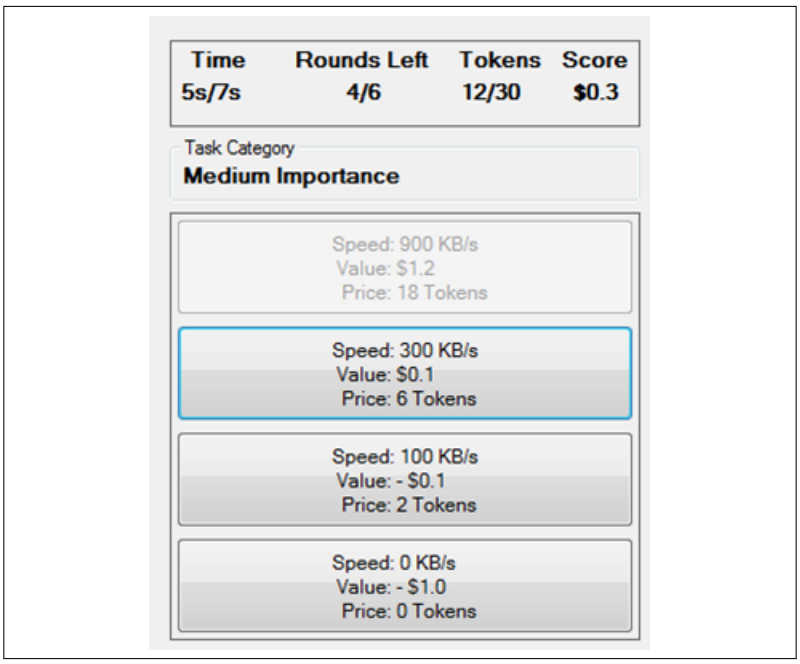


Figure 2: Screenshot of the market game used in the experiment.

amount that is added to a user's *score* when that choice is selected. The third line shows the *price* of that choice in tokens. When the user selects a particular choice, the corresponding number of tokens is subtracted from his current budget. Now let's look at the top of the application. On the far right, it shows the user's score in the current game. After every choice the user makes, the corresponding value of that choice is added to this score which ultimately determines the final game score after the 6th round.

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 towards 0 as the user spends tokens. As a user's budget decreases during a game, those 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 2). 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 2 the user still has 5 seconds left to make a decision in the current round).

Between the information panel at the top and the first choice button is the game's *task category label*. In every round, the user can be in one of three task categories: 1) high importance, 2) medium importance, and 3) low importance (note that this corresponds to the original premise that users are doing tasks of different importance at different points in time). Every round, one of these three categories is chosen randomly with probability 1/3. The task category determines the distribution of the values for the four choices that the user can expect to see in this round. Table 2.2 shows an overview of the values the user can expect in the three categories for a game with 4 choices.[1] As one would expect, selecting the higher speed choices in the "high importance" category gives the user very high value, while choosing low speeds in the high importance category leads to a severe penalty.

[1]Note that the values shown in Table 2.2 are only the averages of the values the user can expect to see. Each valuation is perturbed upwards or downwards with probability 1/3 each, to introduce additional stochasticity in the game, and avoid that the users can memorize a fixed set of values for each task category.

	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

Compare that to the "low importance" category, where the user can earn less value for selecting high speeds, but is also penalized less for selecting the lowest speed.

The user's problem when playing the game is to allocate the budget of 30 tokens optimally over 6 rounds, not knowing which categories with which exact values will come up in future rounds. In some of our experiments, we 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 (out of 3 possible price levels) he will be facing next. This problem constitutes a sequential decision making problem under uncertainty. 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 use the first line on each button to display the speed of that choice to provide each choice with a natural label and to give the UI a little more structure.

2.3 MDP Formulation and Q-Values

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

- **State Space:** *CurrentRound* × *CurrentBudget* × *CurrentCategory* × *CurrentValueVariation* × *CurrentPriceLevel*.
- **Actions:** Each choice affordable in the current round.
- **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*, *CurrentValue-Variation* and *CurrentPriceLevel* transition stochastically.

With six choices and changing prices, the resulting MDP has approximately 1,180,000 states (*Currentround* is between 1 and 6, *CurrentBudget* varies between 30 and 0, *CurrentCategory* varies between 1 and 3, *CurrentValueVariation* varies between 1 and 3^6, denoting for every choice whether the value is perturbed upwards, downwards, or at the normal level, and *CurrentPriceLevel* varies between 1 and 3. In each state there are at most 6 actions possible, thus leading to approximately 7 million state-action pairs. Using dynamic programming, we can solve games of this size relatively quickly (in less than 20 seconds). Thus, we can compute the optimal policy, i.e., we know exactly, for each possible situation that can arise, which choice is currently best according to the optimal MDP-policy. Note that this policy is, of course, computed assuming that the future states are not known; only the transition probabilities as described above are known.

Solving for the optimal MDP-policy 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. We use this concept repeatedly in the analysis section.

2.4 The Quantal-Response Model

A well-known theory from behavioral economics asserts that agents are more likely to take an action the higher its value, or equivalently, users 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 to be:

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

where $Q(a_i)$ denotes the Q-value of action a_i. In this model, the parameter λ is a precision parameter, indicating how sensitive users are to differences between the Q-values. A λ value equal to zero 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 parameter λ that best fits the data. Equipped with such a λ this provides us with a user model which we can use to optimize the UI for behavioral play (see Wright and Leyton-Brown (2010) for a comparison of behavioral models).

3 Experiment Design

In this section we describe in detail the experiment design. Most importantly, this includes a detailed description of the 4 UI design levers that we studied. We then discuss the details of the different treatment variations, our subject pool, and the exact experimental set-up and payment structure.

3.1 The Four Design Levers

In general, a market UI designer has significant freedom in designing both the user interface and aspects of a market for an application. Consider again Figure 2, where we display a screenshot of *one particular version* of the game. In our domain, the design space includes: 1) how many choices do we offer the user, 2) what is the 3G speed of each choice in each situation, and 3) what is the price of each choice in each situation. The only thing we cannot reasonably control as a market UI designer is the value a user has for a choice, because that depends on a user's intrinsic value for speed in a particular moment. In our experimental study, we explore this design space as completely as possible and study the following four design levers:

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

2. **Fixed vs. Changing Prices:** In the *fixed price* treatment, the same choice always costs the same number of tokens (2 tokens per 100KB/s). In the changing price treatment, one of three price levels is chosen randomly with probability 1/3, where the price per 100 KB/s is either 1 token, 2 tokens, or 3 tokens (thus, 500KB/s cost either 5 tokens, 10 tokens, or 15 tokens).

3. **Fixed vs. Adaptive Choice Sets:** In the *fixed choice set* treatment, the users always had 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 choices available to the users varied from round to round, depending on the current category (e.g., in the high category, more high speed choices were available, in the low category, more low speed choices were available).

4. **UI Optimization:** This design lever describes which method is used to determine the composition of the choice sets (i.e., which speed levels are available to the user). In the *optimized for optimal play* treatment, the choice sets are optimized (to maximize the expected score per game) based on the MDP model and assuming optimal play. In the *optimized for sub-optimal play* treatment, the choice sets are optimized assuming behavioral play where actions are chosen according to the quantal-response model.

3.2 Game Complexity and Time Limits

To study the effect of the UI design on a user's ability to make good economic decisions, we need a decision problem with a suitable complexity. If the problem is too easy or too hard, then changes to the UI would likely have no effect. To create a decision problem with just a few choices that is not too easy to solve, we put a fixed time limit on the users' decision, because prior research has shown that users make worse decisions when under time pressure (see, e.g., Gabaix et al. (2006)). With an unlimited amount of time, it shouldn't make a difference whether the user was facing three or six choices in each round. However, under time pressure, coming up with a good strategy might be much harder in a more complex UI than in a simple UI. We used two different time treatments for each user: the first treatment was a fixed time limit per round of 12 seconds. If a user doesn't make a choice within 12 seconds, the lowest choice (with 0KB/s for 0 tokens) is chosen and the game transitions to the next round. The time resets in every round. The second time treatment gave the user 7 seconds per round. In both of these *exogenous* time limit treatments, the game started beeping three seconds before the end of a round to remind the user that the he has to make a decision soon. Note that letting the time run out generally led to the selection of a very bad choice because the lowest choice was rarely a good choice, and always came with a very negative value.

When designing the game, we went through an iterative design process, testing various versions of the game with a group of research interns, until we found the final version of the game as described above. We kept adding more and more stochastic transitions to the game until we could not find any simple heuristic for playing the game well. We calibrated the game (i.e., the size of the budget, the nominal values of the choices, the prices, the number of rounds) in such a way that random play has a highly negative expected score, but that optimal play leads to a score around \$1 on average. 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.

The 7-second and the 12-second time limits were also chosen carefully. In a series of pre-tests we found that for some users, having less than 7 seconds put them under too much time pressure such that they were essentially unable to play the game. 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 such that they could process most of the information available to them.

Number Of Choices	12-second game	7-second game
3	4 ×	4 ×
4	4 ×	4 ×
5	4 ×	4 ×
6	4 ×	4 ×

Table 2: Design of Experiment 1. Each participant played 32 games. The design lever *Number of Choices* was a within-subject factor, the design lever *Fixed vs. Changing Prices* was a between-subjects factor.

3.3 Methodology and Experimental Set-up

We recruited 53 participants (27 males, 26 females) 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. The median age of our participants was 39, ranging from 22 to 54. None of the participants worked for the same company, but all of them had some familiarity with smartphone interfaces. We ran one participant at a time with each session lasting about 1.5 hours. The users filled out a pre-study questionnaire (5 minutes), went through a training session where the researcher first explained all the details of the game and then gave the participants the opportunity to play 12 training games (20 minutes), participated in the experiment (55 minutes) and then completed a post-study survey (10 minutes). We ran the software on a single 3 GHZ Dell computer at full screen resolution. The participants were compensated for their participation in the study 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 around $1. After each game, we show the user his score from the last game and the accumulated score over all games played so far. [2] The final giftcard amounts of the 53 users varied between $4.60 and $43.70, with a median amount of $24.90.

3.4 Treatments

The study was split into two separate experiments. In Experiment 1 we had 35 out of the 53 participants, and we tested the design levers *Number of Choices* and *Fixed vs. Changing Prices*. *Number of Choices* was a within-subject factor, and *Fixed vs. Changing Prices* was a between-subject factor. We had 18 participants who only played games with fixed prices, and 17 who only played games with changing prices. Table 2 depicts the experiment design for each individual user. For each treatment, each user played four games with the 12-second time limit and four games with the 7-second time limit.[3] 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 and then played the four

[2] Note that we had originally 56 participants in our study, but we had to exclude 3 participants from the first experiment (2 males, 1 female) because they did not understand the game well enough and achieved a negative overall score.

[3] Each user also played another game for each treatment with an overall time limit of 4 minutes. The analysis of those endogenous time games is still underway.

7-second games. Thus, every participant played 32 games with 6 rounds each, which gives us a data set with a total of 1,120 games or 6,720 rounds from Experiment 1.

In Experiment 2 we had 18 participants and we tested the design levers *Fixed vs. Adaptive Choice Sets* and *UI Optimization*, and both were within-subjects factors. See Table 3 for a depiction of the experiment design for each individual participant. We randomized the order of the 4 different treatments. As before, for every treatment, every user played 4 12-second games and 4 7-second games. Every participant played 32 games with 6 rounds each which gives us a data set with 576 games or 3456 rounds. Thus, from both experiments together, we obtained more than 10,000 data points, where each data point corresponds to a decision that a participant made in one particular game situation.

Treament	12-second game	7-second game
Fixed-Choice-Sets & Optimized-For-Opt	4 ×	4 ×
Adaptive-Choice-Sets & Optimized-For-Opt	4 ×	4 ×
Fixed-Choice-Sets & Optimized-For-SubOpt	4 ×	4 ×
Adaptive-Choice-Sets & Optimized-For-Sub-Opt	4 ×	4 ×

Table 3: Design of Experiment 2. Every participant played 32 games. Both design levers *Fixed vs. Adaptive Choice Sets* and *UI Optimization* were within-subject factors.

4 Analysis and Results

In this section we present a detailed statistical analysis of the experimental data obtained from both experiments.

4.1 Choice of Regression Models

Using multiple (repeated) measurements from individual users violates the independence assumption of standard (OLS or logistic) regression models, because multiple measurements from the same user are not independent from each other. That is why for all of the statistical analysis of the data we use *Generalized Estimating Equations (GEE)*, an extension of generalized linear models (Nelder and Wedderburn (1972)) that allows for the analysis of repeated measures or otherwise correlated observations. When analyzing binary decisions (e.g., did the user make the optimal choice) we use the logit link function and the binomial distribution (as in logistic regression). When analyzing scale variables like *value loss*, *efficiency* or *decision time*, we use the identity link function and the Normal distribution (as in linear regression).

To compare the *goodness of fit* of different models, GEE provides the QIC and QICC information criteria (Nelder and Wedderburn (1972)) which are based on a generalization of the likelihood (comparable to R^2 in OLS regressions, however, here smaller values denote a better fit). We use the QIC value to choose between different correlation structures, and the QICC value to choose between

different models (i.e., sets of model terms). We tested a series of correlation structures, including *compound symmetry* and *unstructured*. However, assuming independence led to the smallest QIC values, indicating the best fit. Thus, we always report the results using generalized estimating equations that assume independence. Note that GEE has the nice property that even if the correlation structure is misspecified, the coefficient estimates are still consistent (but may have larger standard errors), which makes using GEE particularly attractive. With the independence assumption, GEE can be seen as an extension of (logistic/linear) regression methods for clustered data. Note that there is no widely accepted definition of standardized coefficients for a logistic regression model. Thus, when reporting regression results using the logit link function, we only report the non-standardized coefficient estimates B and the corresponding odds ratios $Exp(B)$. Thus, when interpreting the results, we always have to take the standard deviation of the corresponding predictor into account. When reporting results using the identity link function (linear regression), we also report the standardized coefficients.

In analyzing the data, our general goal is to understand which factors influence whether users make good or bad choices. There are two ways we can look at the data. First, we can look at the results of the *games*, measure the average efficiency that users achieved per game, and compare how efficiency differed under different treatments. Second, we can look at the individual *rounds* of each game, and measure whether users chose the optimal action or not, and which factors influenced their performance. Analyzing the individual rounds gives us a more detailed look at what actually happened, because we can take factors into account that change every round, like the Q-value differences, number of choices left, position of the optimal choice, value of the optimal choice, budget, time, etc. Thus, we begin our analysis by taking a very close look at the individual rounds, before moving on to the analysis of the games.

The actions available to a user in each round have an inherent order based on their Q-values, and we can rank them from best to worst. Thus, in the most general model, the dependent variable of the regression model would be the rank of the chosen action. *Ordered logistic regression* is a suitable regression model for this case. However, this model can only be used when the *proportional odds assumption* is satisfied, which says that the relationship between all pairs of outcome groups (i.e., values of the dependent ordinal variable) is the same. This assumption is clearly violated in our domain. For example, it makes sense that the Q-value difference between the best and second best action is very predictive for whether a user chooses the best or second best action, but it isn't for whether the user chooses the second best or third best action. There is a generalization of the ordered logistic regression model called the generalized ordered logit model, but this essentially builds a separate model for each pair of outcomes, which makes interpreting the results very difficult. However, we are mainly interested in understanding when the user is able to find the *optimal* choice. Furthermore, the best and second best choices make up the majority of outcomes (ranging from 70% for the game with 6 choices, to 98% for the game with 3 choices). Thus, we simplify the analysis of the round-based data, and study the binary dependent variable *OptChoice*, which is 1 if the user clicked on the optimal choice, and 0 otherwise.

4.2 Behavioral Results

Data Selection: From both experiments together, we obtained 10,176 data points. Because we tested four different design levers, there is a lot of variance in the data. For this first analysis, to most cleanly identify the behavioral factors unrelated to the four design levers, we only consider the data points from Experiment 1 with fixed prices, 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 17 cases where only one or two choices were left, which leaves us with 2,863 data points.[4] 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., rounds). Note that we consider games with a 7-second and with a 12-second time limit, because we could not find a statically significant effect of the time limit on decision performance.

Factors	(1)		(2)		(3)	
	B	Exp(B)	B	Exp(B)	B	Exp(B)
Intercept	-0.816**** (0.1408)	0.442****	-1.529**** (0.1593)	0.217****	-1.398**** (0.1657)	0.247****
Lambda	0.150**** (0.0180)	1.162****	0.161**** (0.0197)	1.175****	0.151**** (0.0176)	1.163****
QvalueDiff			5.868**** (0.4353)	353.713****	5.884**** (0.4358)	359.392****
female?					-0.130* (0.0716)	0.878*
Fit (QICC)	(3771.953)		(3589.063.360)		(3588.483)	

Table 4: GEE for dependent variable *OptChoice*. 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. N=2756.

The Quantal Response Model: As a first step, we test whether the quantal response model is a good model for user behavior in our experiment, and whether the individual users exhibit significant differences in their play. We compute a separate maximum-likelihood parameter λ_i for each user i in the data set. This parameter can be seen as measuring how "rational" a user's play was. It turns out that the users exhibited large differences, with a minimum λ of 3.9, a maximum of 9.0, and a median of 6.8. In this subset of the data, this translated to payments between \$4.60 and \$34.00, and the correlation between λ and the final payment was 0.68, i.e., very high. Now consider Table 4 which presents the results from fitting GEE with *OptChoice* as the dependent variable. In column (1), we see that the parameter *lambda* has a statistically significant effect on the user's likelihood for choosing the optimal choice. 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 $\lambda = x - 1$. As we add more factors to the regression, we will see that this effect is very robust and remains statistically significant. Thus, we always control for lambda as a way to control for a user's individual "rationality".

Q-Value Differences: Note that the λ-parameters are measures across all time steps and for all different game situations. Thus, they are a very general measure of a user's degree of rationality. We now look more directly at the effect of the Q-values for each individual action by adding the factor

[4]With one choice left, there was nothing for the user to decide (and we only had 7 data points withe one choice left). Having only 2 choices left was also a very unusual decision situation, usually towards the end of a game when a user was running out of budget, or when he has previously made a mistake. For the data set we consider here, we only had 10 data points where 2 choices were left.

Factors/Covariates	(1)		(2)	
	B	Beta	B	Beta
Intercept	0.138**** (0.0118)		0.231**** (0.0154)	
Lambda	-0.11**** (0.0015)	-0.141****	-0.013**** (0.0021)	-0.151****
female?	-0.004 (0.0032)	-0.018	- 0.016*** (0.0056)	-0.066***
Goodness of Fit (QICC)	36.302		24.026	
Cases Considered	All (N=2756)		OptChoice=0 (N=1246)	

Table 5: GEE for the dependent variable *ExpectedValueLostFromThisChoice*. 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.

QvalueDiff, the difference between the Q-values of the best and second-best action to the regression. In column (2) in Table 4 we see that the Q-value difference is highly statistically significant and has an odds ratio of 353. This is the odds ratio for a one unit change in the Q-value difference. In our data, the Q-value difference varies between 0 and 0.84, with a mean of 0.11. The odds ratio for a change of 0.1 is 1.798. 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%. This is a very large effect, and we will see that it is robust to adding more factors to the regression.

Age: Next we test whether users' performance differed by age. In this data sample, our participants were between 24 and 54, with a median age of 40. However, adding the factor *Age* to the regression, we did not find a statistically significant effect on the dependent variable, and thus we leave it out for the remaining analyses.

Male vs. Female Users: Prior research in psychology and human computer interaction has established significant gender differences in various cognitive tasks, and shown that men and women uses different strategies and excel in different environments (see, e.g., Czerwinski et al. (2002S)). This motivated us to test if there were significant gender differences in our experiment as well. In column (4) of Table 4 we see that indeed, there is a small, but statistically significant effect. The female participants were less likely to choose the optimal action. In particular, their odds were 12% lower than the odds for men. We will see later, that this gender effect is robust, in size and statistical significance. However, this is not the end of the story, because it only says that the female participants chose a sub-optimal action more often, but not which one. Consider now Table 5 where we present the results from running a linear regression where the dependent variable is *ExpectedValueLostFromThisChoice*. In every round of every game, this variable equals zero if the user chose the optimal choice, and it is equal to the difference between the Q-value of the optimal choice and the Q-value of the choice that the user selected. Thus, it is a (probabilistic) measure for how much a user is expected to lose (over the course of the rest of the particular game) due to one sub-optimal choice. In that sense, it is a proxy for efficiency, but with lower variance and one that we can measure every round.

Now consider column (1) of Table 5 where we ran the regression with *Lambda* and *Female* as factors. We can see that there is no statistically significant gender effect on the expected value lost, which

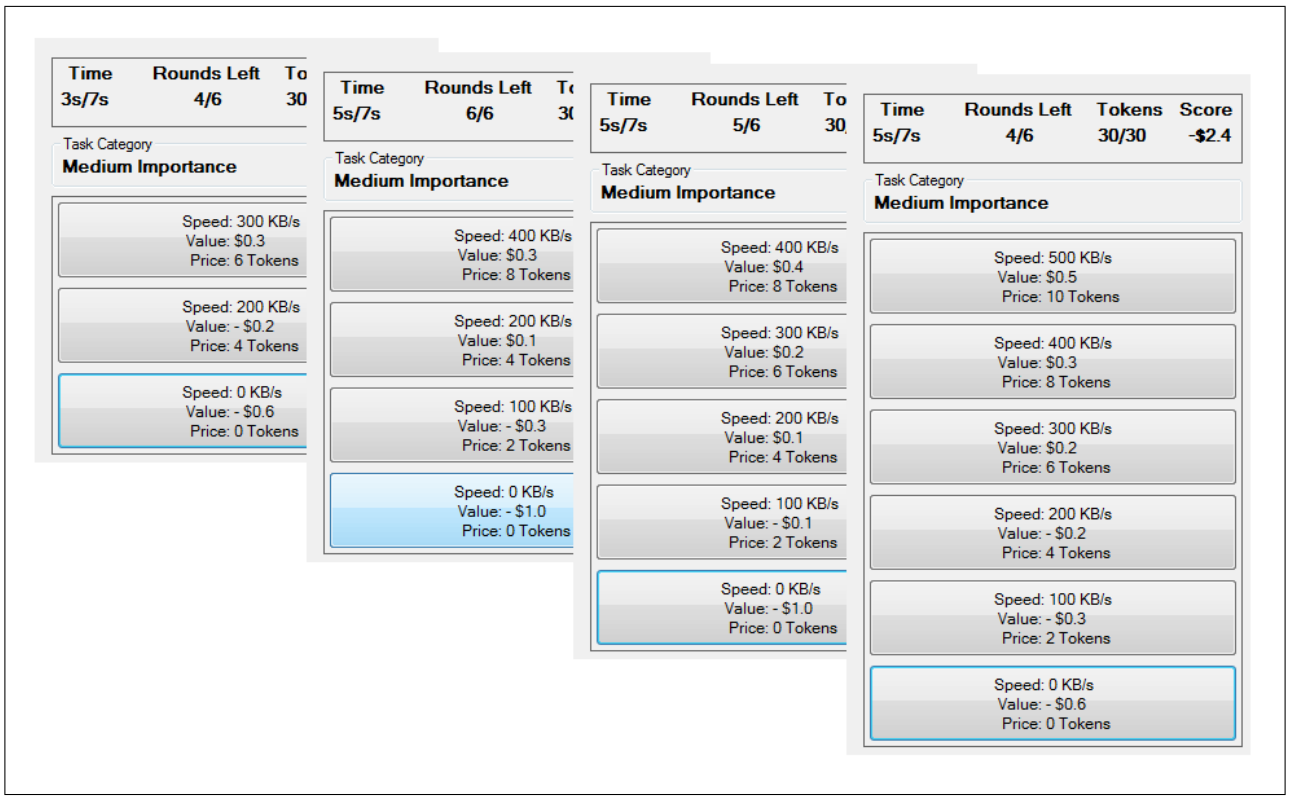


Figure 3: Different screenshots for games with 3, 4, 5, and 6 choices. All screenshots are for the "medium importance" category, however, the values are also randomly perturbed upwards or downwards.

corresponds to the finding we will present later, that men and women do equally well in terms of efficiency. Now consider column (2) of Table 5 where we ran the same regression, but only for those cases (i.e., rounds) where the user chose a sub-optimal action. Now we see that the factor *Female* has a negative coefficient and is highly statistically significant ($p < 0.01$). This shows that, while female participants make more mistakes, the mistakes they make are less severe than then ones that men make when they make mistakes.

UI Design and Number Of Choices: We now move on to the analysis of how the UI design affects the users' performance in making optimal choices. In particular, we analyze the effect of varying the number of choices available to users. Consider Figure 3 where we display screenshots of the 4 different types of games each user played, with 3, 4, 5 and 6 choices (we randomized the order in which the users played those games). Now consider Table 6 where we continue the regression analysis for the dependent variable *OptChoice*. We control for the factors that we already found to have statistically significant effects, namely *Lambda*, *QvalueDiff*, and *Female*. In column (1), we add *numChoices* to the regression, representing the type of game the user was playing (i.e., with 3,4,5 or 6 choices). We see that the factor has a large and highly statistically negative effect on *OptChoice*. Holding all other factors constant, increasing the number of choices by 1 reduces the odds for making the optimal choice by 32%. This is the effect that we had expected: a more complex UI (e.g., more choices) makes it harder for the users to find the optimal choice. This suggests, that we can *potentially* improve users'

Factors/Covariates	(1)		(2)	
	B	Exp(B)	B	Exp(B)
Intercept	0.413* (0.2503)	1.511*	0.229 (0.2575)	1.257
Lambda	0.157**** (0.0179)	1.170****	0.158**** (0.0176)	1.171****
QvalueDiff	5.075**** (0.4306)	159.959****	4.883**** (0.4200)	132.015****
female?	-0.128* (0.0750)	0.880*	-0.128* (0.0747)	0.880*
numChoices	-0.391**** (0.0449)	0.677****		
numChoicesLeft			-0.355**** (0.0451)	0.701****
Goodness of Fit (QICC)	3475.661		3499.991	

Table 6: GEE for dependent variable *OptChoice* studying UI complexity in terms of number of choices. 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.

overall performance, by providing them with fewer instead of more choices. Now consider column (2) of Table 6 where we have removed *numChoices* from the regression, and added *numChoicesLeft*. The difference between these two factors is that *numChoicesLeft* does not remain constant during a game, but always denotes how many choices the user still has left, given the price of the current choices and his current budget. For example, in a game with 6 choices, as the user continues spending his budget, *numChoicesLeft* will keep decreasing monotonically until the last time step. We see that *numChoicesLeft* has a similarly large negative effect on *OptChoice* and is also highly statically significant.

Obviously, *numChoices* and *numChoicesLeft* are positively correlated, i.e., if *numChoices* is large, then *numChoicesLeft* is more likely to be large as well. In column (1) of Table 7 we add *numChoices* back into the regression, and we see that now that we are controlling for *numChoices* the factor *numChoicesLeft* is no longer statistically significant. This suggests that *numChoices* is the important factor, and *numChoicesLeft* only shows up as statistically significant, because of its correlation with *numChoices*. In fact, in additional analyses, we haven't found *numChoicesLeft* to have a statistically significant effect, when looking at fixed values of *numChoices*. Thus, going forward, we will keep *numChoices* in the regression to control for the UI complexity effect, but we will leave *numChoicesLeft* out of the regression.

Incomplete Search and Position Effects: By design, the game exhibits a strong ordering effect: the value of the choices decrease monotonically from top to bottom, as do the prices. It is conceivable, that users scan the choices in a linear way, either from top to bottom or from the bottom to the 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, e.g., Dumais et al. (2010); Buscher et al.

Factors/Covariates	(1)	
	B	Exp(B)
Intercept	0.362 (0.2616)	1.436
Lambda	0.156**** (0.0179)	1.169****
QvalueDiff	5.206**** (0.4253)	182.395****
female?	-0.128* (0.0748)	0.880*
numChoices	-0.502**** (0.1090)	0.606****
numChoicesLeft	0.123 (0.1158)	1.131
Goodness of Fit (QICC)	3476.190	

Table 7: GEE for dependent variable *OptChoice* studying UI complexity, controlling for both, the total number of choices, and the number of choices left. 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.

(2010). For example, it could be that users are always more likely to click on a choice towards the top rather than towards the bottom, no matter how many choices there are, or what the values and prices of those choices. Fortunately, we can control for positional effects by adding information about the position (or rank) of the optimal choice to the regression. Consider column (1) in Table 8 where we added the control variable *optRelativeRank* to the regression. The variable denotes the "relative rank" or "relative position" of the optimal choice, taking into account the currently unavailable choices. For example, consider a game with 6 choices. If there are currently 4 choices left and the optimal choice is the third from the top, then the absolute position of that choice would be 2 (we start counting at 0 from the top), but the relative rank is 0. We use the relative rank rather than the absolute rank for two reasons. First, using the absolute position of the choice 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 regression. However, we have also performed the same analyses with the absolute position control variable, and obtained qualitatively similar results.

In column (1) of Table 8 we see that *optRelativeRank* has a very strong, and highly statistically significant negative effect on *OptChoice*. Note that rank 0 is at the top, and all coefficient estimates are relative to *optRelativeRank=0*. We see that the lower the rank of the optimal choice, the less likely were the users to choose the optimal action. As we go from rank=0 to rank=5, the coefficients decrease monotonically, and except for *optRelativeRank=1*, all of the effects are highly statistically significant. Especially for the very low ranks, the effect on *optChoice* is very strong. 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 85%, and if *optRelativeRank=5*, the odds decrease by 98%. Thus, the position effect is indeed very strong and we need to control for it. Note that the other factors we are controlling for are still statistically significant and the coefficients are relatively

Factors/Covariates	(1)		(2)		(3)	
	B	Exp(B)	B	Exp(B)	B	Exp(B)
Intercept	-0.341 (0.2664)	0.711	-0.339 (0.2584)	0.713	-0.439* (0.2558)	0.645*
Lambda	0.150**** (0.0189)	1.162****	0.150**** (0.0188)	1-162****	0.145**** (0.0197)	1.156****
QvalueDiff	4.428**** (0.5060)	83.741****	4.427**** (0.5039)	83.671****	4.599**** (0.4998)	99.387****
female?	-0.151** (0.0687)	0.860**	-0.151** (0.0695)	0.860**	-0.166** (0.0734)	0.847**
numChoices	-0.086* (0.0486)	.917*	-0.087* (0.0513)	0.917*	-0.065 (0.0584)	0.937
optRelativeRank=5	-3.884**** (0.9824)	0.021****	-3.881**** (0.9925)	0.021****	-4.068**** (1.0438)	0.017****
optRelativeRank=4	-1.902**** (0.4482)	0.149****	-1.900**** (0.4594)	0.150****	-1.853**** (0.4948)	0.157****
optRelativeRank=3	-1.205**** (0.2692)	0.300****	-1.203**** (0.2974)	0.300****	-1.183**** (0.3372)	0.306****
optRelativeRank=2	-0.619** (0.2784)	0539**	-0.617** (0.2967)	0.539**	-0.523 (0.3322)	0.593
optRelativeRank=1	-0.169 (0.2272)	0.845	-0.168 (0.2358)	0.845	-0.178 (0.2493)	0.837
optRelativeRank=0	0	1	0	1	0	1
optimalChoiceNegative?			-0.002 (0.0896)	0.998	-1.314**** (0.2270)	0.269****
currentCategory=2					1.539**** (0.2088)	4.658****
currentCategory=1					0.032 (0.1282)	1.033
currentCategory=0					0	1
Goodness of Fit (QICC)	3343.975		3345.975		3286.565	

Table 8: GEE for dependent variable *OptChoice* studying 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.

stable, which makes sense, given that there are no correlations between them and *optRelativeRank*.

Loss Aversion: Controlling for the position effect is particularly important when analyzing the effect of the nominal value and price of the optimal choice. We now consider if it makes a difference whether the optimal choice has a positive or negative (short-term) value. Of course, for a fully rational player, that shouldn't matter. It is inherent to our game that the optimal strategy sometimes requires taking a short-term loss, for larger gains in a later round. With a limited budget of 30 tokens, the user cannot always afford to select choices with positive values (see Figure 3).

However, loss-aversion is a well-known effect in behavioral economics, and thus we expected to find it in our data as well. Now consider column (2) of Table 8 where we added *OptimalChoiceNegative?* to the regression, an indicator variable that is 1 when the nominal value of the optimal choice is negative, and 0 otherwise. We see that this factor does not show up has having a statistically significant effect

on *Optchoice*. However, it turns out that *OptimalChoiceNegative?* does in fact have a strong effect, but only in certain game situations.

Remember that the distribution of values changes randomly. The three categories "high", "medium", and "low" give a rough indication for the distribution of values for all choices, but in addition, each individual value is also randomly perturbed upwards or downwards. By taking a closer look at the distribution of values in the different categories, we gain a better understanding of when *OptimalChoiceNegative?* can have an effect. For example, in the *low* category, often times all of the choices have a negative value, or at least the first and second best choice do. In such game situations, *OptimalChoiceNegative?* cannot have an effect on *OptChoice*. Consider now column (3) of Table 8, where we added the factor *CurrentCategory* to the regression. Now, two things happen. First, *optimalChoiceNegative?* now has a large negative coefficient, and is highly statistically significant. This suggests that once we are controlling for the distribution of the values, holding everything else constant, it makes a large difference in users' play, whether the optimal choice has a positive or negative value, providing strong evidence for our loss aversion hypothesis.[5] The second effect we see is that while there is no statistically significant difference between categories 0 and 1, *CurrentCategory=2* has a large positive coefficient and is highly statistically significant. This is surprising, at first, because it is unclear why the choice problem should be much easier just because all values are relatively low. However, it turns out that most of this effect can be explained by the interaction of *CurrentCategory* and *optimalChoiceNegative?* (i.e., in category 2 all choices will often times have a negative value).

To get a better understanding of the loss aversion effect, we looked at two interaction effects. First, the previous analysis already suggests that there is an interaction between *optimalChoiceNegative?* and *CurrentCategory*. Second, we hypothesized that it also makes a big difference for loss aversion, whether the choice one position higher than the optimal choice also has a negative value, or whether that choice has a positive value. Thus, we also consider the interaction effect with *OneHigherNegative*. Now consider column (1) of Table 9 where we added 8 indicator variables to study the combined interaction effects of *OptimalChoiceNegative*, *oneHigherNegative* and *currentCategory*. Note that all effects of the indicator variables are relative to the default case where *CurrentCategory=0* and both the optimal choice, and the one above it, have positive values. The first thing we see is that, when both the optimal choice and one above it are both positive, then there is no statistically significant effect of *CurrentCategory*. In a separate analysis, we also looked at the effect of *CurrentCategory* when the optimal choice has a negative value, and there was also no statistically significant effect. Thus, this provides evidence for our intuition that the game is not more or less difficult just because the value distribution is shifted upwards or downwards.

Now, let's take a closer look at *OptimalChoiceNegative=1*. First, we see that there is no statistically significant effect when *CurrentCategory=0* and when *CurrentCategory=2*. A closer investigation of this (not shown here) reveals that for *CurrentCategory=0*, the optimal choice is almost never negative, and thus there are simply too few data points for *OptimalChoiceNegative=1*. For *CurrentCategory=2*, the optimal choice is almost never positive, and thus there are too few data points with *OptimalChoiceNegative=0*. This leaves *CurrentCategory=1*, where we indeed see a large and statistically significant negative effect of *OptimalChoiceNegative=1* on *Optchoice*. Furthermore, by looking at the

[5]This loss aversion behavior exhibited by our users obviously represents erroneous, and sub-optimal behavior. This kind of behavior may have severe consequences in a many real-world environments. For example, consider those people living from paycheck to paycheck, i.e., people having to make sequential decisions on a fixed budget. If they forego big wins in the future to avoid small losses now, this significantly impacts their utility. Note, however, that while it is easy for us to compute the optimal strategy in our domain, it us unclear what other effects in terms of *ease of justification* and *avoidance of negative emotions* loss-averse behavior implies (see Payne and Bettman (2001) for more on this topic).

Factors/Covariates	(1)	
	B	Exp(B)
Intercept	-0.433* (0.2514)	0.648*
Lambda	0.144**** (0.0203)	1.155****
QvalueDiff	4.605**** (0.4918)	100.016****
female?	-0.174** (0.0763)	0.840**
numChoices	-0.066 (0.0581)	0.936
optRelativeRank=5	-4.086**** (1.0302)	0.017****
optRelativeRank=4	-1.846**** (0.4798)	0.158****
optRelativeRank=3	-1.186**** (0.3292)	0.305****
optRelativeRank=2	-0.531 (0.3342)	0.588
optRelativeRank=1	-0.186 (0.2476)	0.831
optRelativeRank=0	0	1
[optimalChoiceNegative=1 × oneHigherNegative=1 × currentCategory=2	0.248** (0.1255)	1.281**
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=2	0.070 (0.4076)	1.073
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=2]	-1.199 (1.7347)	0.301
[optimalChoiceNegative=1 × oneHigherNegative=1 × currentCategory=1]	-1.038*** (0.4043)	0.354***
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=1]	-1.575**** (0.3256)	0.207****
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=1]	0.066 (0.1327)	1.068
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=0]	-0.322 (0.6351)	0.725
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=0]	0	1
Goodness of Fit (QICC)	3287.205	

Table 9: GEE for dependent variable *OptChoice* studying loss aversion with interaction effects. 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.

interaction with *oneHigherNegative*, we see that the negative effect of *OptimalChoiceNegative=1* is particularly strong when *oneHigherNegative=0* which concurs with our hypothesis, i.e., users are more likely to make a mistake when the optimal choice has a negative value and when the choice right above it has a positive value. When only the optimal choice is negative, this leads to a reduction of 65% in the odds for getting the optimal choice right (compared to the default case). When in addition, the choice right above has a *positive* value, then the odds are reduced by another 15% points, such that total reduction in the odds is almost 80%. 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 still part of the regression and we are thus already controlling for the position effect. There is a third interaction effect that shows up as statistically significant, namely when both the optimal choice and then one above it have a negative value in category 2. At this point, however, we do not have an explanation for the origin of this effect. As mentioned above, a separate analysis showed no statistically significant effect of the categories by themselves.

The Role of Time, Budgeting, and Learning: In Table 10, we added four additional covariates at once: the number of choices left in the game, the current time step (between 1 and 6), the user's current budget (in tokens), and the *gameCounter*, indicating how many games a user has already played. We are mainly interested in the effect of time, within a game, and over the course of the whole experiment. We added *NumChoicesLeft*, *CurrentTimeStep*, and *CurrentBudget* to the the regression simultaneously, because they are correlated with each other in a very intricate way. As the game progresses, the variable *CurrentTimeStep* increases, the user spends more and more of his budget, and thus the choices that are left available to him decrease. Furthermore, the variable *NumChoicesLeft* is also correlated to *NumChoices*. We see in Table 10, that once we have added all of these factors to the regression, none of them show up as statistically significant. We have also tried adding them to the regression one by one, and we have analyzed different subsets of the data to remove some of the interaction effects. However, we could not find evidence that these factors have a statistically significant effect in any direction, when controlling for all other variables.

Note that in the last column of Table 10 we also added a variable *GameCounter* denoting the number of games a user had already played when making the current decision. The goals is to control for learning effects over the course of the experiment. However, we did not find any statistically significant effect. Note that all participants went through an extensive training period before the experiment itself started where they had the opportunity to play 12 different games. It seems that the training period was long enough to remove any additional learning effects.

Review of Behavioral Effects: Before moving on to the efficiency analysis, let's briefly review the main findings of this section. We saw that *Lambda* has a large, statistically significant effect, i.e., there are significant differences in individual users' decision making performance. Second, *QValueDiff* is highly statistically significant, showing that the difference in Q-values between the best and second-best choice is an important factor. Third, we saw that female users miss the optimal choice more often, but that this is counterbalanced by the fact that male users make worse mistakes, losing more value, when they miss the optimal choice. Fourth, we saw that *numChoices* has a large, statistically significant effect, showing that the UI complexity in terms of the number of choices is important. Fifth, we saw that there is a strong position effect, with users selecting the optimal choice more often when its relative rank is high rather than low. Finally, we found a strong loss aversion effect, i.e., users are more likely to miss the optimal choice when its absolute value is negative, in particular when the value of the choice right above is positive.

Factors/Covariates	(1)	
Intercept	0.155 (0.8078)	1.167
Lambda	0.146**** (0.0202)	1.158****
QvalueDiff	5.192**** (0.5134)	179.817****
female?	-0.168** (0.0779)	0.845**
numChoices	-0.216 (0.1609)	0.806
optRelativeRank=5	-4.347**** (1.0569)	0.013****
optRelativeRank=4	-2.042**** (0.5561)	0.130****
optRelativeRank=3	-1.314**** (0.3671)	0.269****
optRelativeRank=2	-0.581* (0.3522)	0.560*
optRelativeRank=1	-0.204 (0.2524)	0.816
optRelativeRank=0	0	1
[optimalChoiceNegative=1 × oneHigherNegative=1 × currentCategory=2	0.345** (0.1450)	1.412**
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=2	0.156 (0.4025)	1.168
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=2]	-0.982 (1.7479)	0.375
[optimalChoiceNegative=1 × oneHigherNegative=1 × currentCategory=1]	-0.834*** (0.4161)	0.434***
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=1]	-1.411**** (0.3193)	0.244****
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=1]	0.094 (0.1397)	1.099
[optimalChoiceNegative=1 × oneHigherNegative=0 × currentCategory=0]	0.205 (0.6549)	1.227
[optimalChoiceNegative=0 × oneHigherNegative=0 × currentCategory=0]	0	1
numChoicesLeft	.194 (0.1741)	1.214
currentTimeStep	-0.194 (0.1337)	0.824
currentBudget	-0.013 (0.0271)	0.987
GameCounter	-0.001 (0.0034)	0.999
Goodness of Fit (QICC)	3270.488	

Table 10: GEE for dependent variable *OptChoice* studying the role of time, budgeting, and learning. 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.

4.3 Efficiency Results

4.3.1 Optimal Efficiency vs. Realized Efficiency

We now transition from the analysis of the users' decisions in individual rounds, to the analysis of their overall performance. Thus, we now study the effect of the individual design levers on the average efficiency that users achieved per game. We could have used the aggregated scores that users achieved per class of game as the efficiency measure. However, that measure was very noisy due to the high degree of randomness in the game itself. To account for this, we computed a different measure of efficiency, removing the randomness as much as possible. First, for each game, we add up the differences between the Q-value of the optimal choice in each round and the choice selected by the user, which gives us the *ExpectedValueLoss* for a game, a probabilistic measure of how much value a user playing a particular strategy would lose in this game on average (thus, also removing the randomness due to cases where the user just got lucky). Second, every game has an *ExpectedOptimalValue*, or optimal efficiency, which is the expected a priori value for playing the game optimally, without knowing the realization of the state uncertainties. This is simply the value of the corresponding MDP. Additionally, every game actually played also has an *OptimalScore* which is the score an optimal player could have achieved in this particular game, had he followed the optimal policy (not knowing the future). Of course, averaged over many games, *OptimalScore* equals *ExpectedOptimalValue*. However, in a particular game, *OptimalScore* can be much higher or much lower than *ExpectedOptimalValue* because of the randomness in the game (e.g., lots of high value choices, or lots of low prices). Thus, we scale each game's *ExpectedValueLoss* by the ratio of *OptimalScore* and *ExpectedOptimalValue* to get a normalized measure for value loss. Then we subtract this normalized measure from the *ExpectedOptimalValue* of the game, to get a measure for *Realized Efficiency*. Note that, if we let the number of games played go to infinity, the regular game scores would approach *Realized Efficiency*. However, with just a few hundred games played per design lever, the impact of the game's randomness on the regular score is too large, which is why we use *Realized Efficiency* instead.

4.3.2 Computational Search for the Optimal UI

In the following section, we consider the data from Experiment 1 and study the effects of changing the number of choices and the effect of having fixed vs. changing price levels on the user's *Realized Efficiency*. When varying the number of choices available to the users from 3 to 6, this still leaves open the question of *which particular choices* to offer the users (i.e., which speed levels). The only constraint we imposed was that the 0KB/s choice had to be included, because that was the only choice with a price of 0 tokens, which had to be available when the user ran out of tokens. For our experiment, we always chose the "optimal" game for each design constraint, where optimal here means highest *ExpectedOptimalValue*. In practice, we wrote a search algorithm that took as input the design parameters (here, number of choices and fixed vs. changing prices), iterated through all possible combinations of choices (i.e., all possible speed level combinations), for each combination solved the resulting MDP to determine its *ExpectedOptimalValue*, and output the design with the highest *ExpectedOptimalValue*. Consider Figure 3 where we display the four designs that our algorithm found for the different number of choices with fixed prices. Note that going from 3 to 4 choices, the algorithm takes out the 300KB/s choice, and instead adds a 100KB/s choice and a 400KB/s choice, because that combination of available choices lead to a higher expected value of the corresponding MDP. Using this method of finding the optimal UI, we guarantee that for every particular set of design criteria, we always present the user with the best possible UI given these constraints. Note that in this section "best-possible" means optimized assuming a perfectly-rational, or optimal, player.

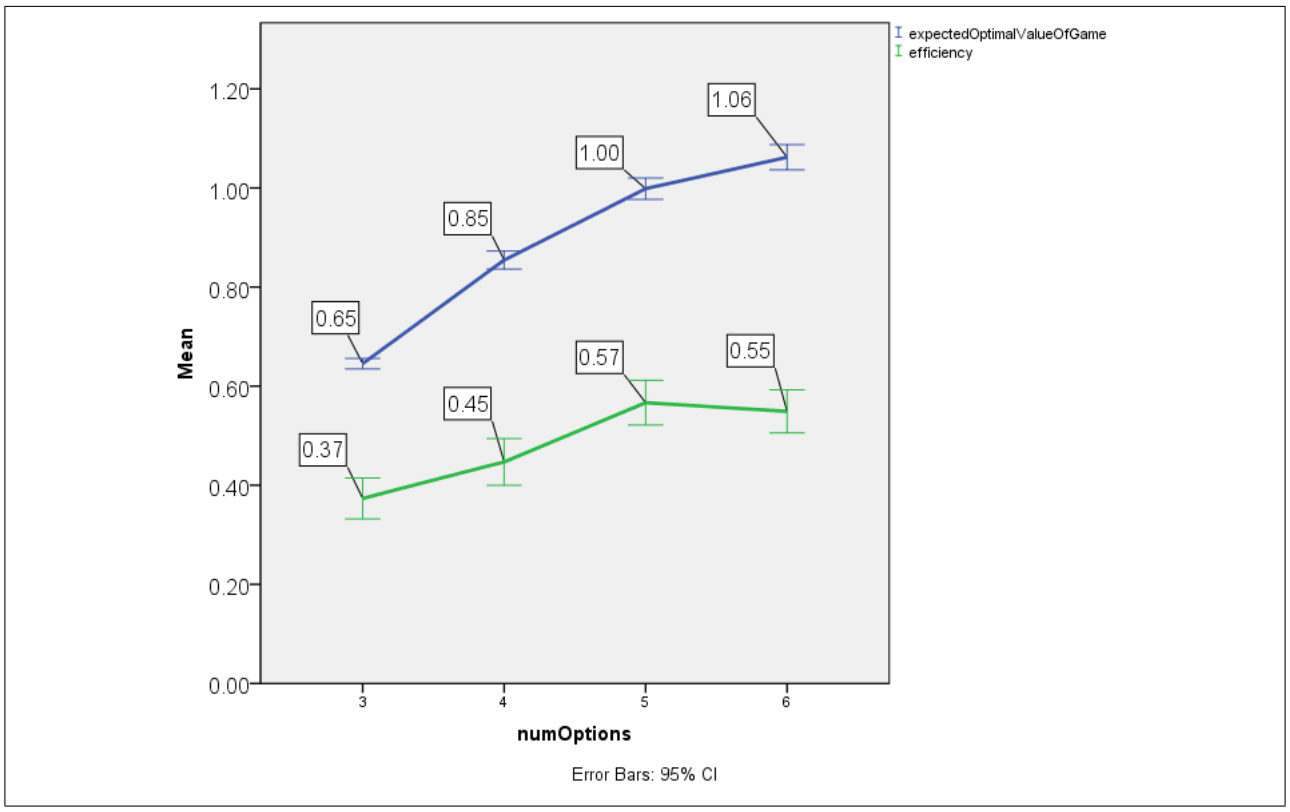


Figure 4: Efficiency for 3, 4, 5 and 6 Choices. The blue line (on the top) correspond to optimal efficiency and the green line (on the bottom) corresponds to the users' Realized Efficiency.

4.3.3 Results for Design Levers 1+2: Number of Choices and Fixed vs. Changing Prices

Regarding the number of choices, our hypothesis was that the users' *Realized Efficiency* first increases as we increase the number of choices, but then peaks at some point, stops increasing further, and then decreases again. We have already seen in the previous section that users make more mistakes in games with more choices (see Table 6). Thus, on the one side, a higher number of choices makes the game more difficult to play. On the other side, the games with a larger number of choices have higher *optimal efficiency* under *perfectly rational play*. Now, consider Figure 4 where we display efficiency results for 3, 4, 5 and 6 choices. While the top line, i.e., *optimal efficiency*, monotonically increases as the number of choices is increased, the bottom line, representing *Realized Efficiency*, only increases as we go from 3 to 4 to 5 choices, but then slightly decreases as we go from 5 to 6 choices. Thus, the disadvantage from adding cognitive load as we go from 5 to 6 choices definitely outweighs the possible benefits of having one more choice available. However, it is unclear if the efficiency only plateaus, or if it actually decreases by a statistically significant amount. Notice that the error bars are relatively large, and in particular the error bars for 5 and 6 choices overlap to a large degree. Thus, we now turn to the statistical data analysis to see if there was a statistically significant decrease in efficiency or not.

Notice that the games with changing prices had a higher *optimal efficiency* than the games with fixed prices, and thus it is important to add this variable to the analysis from the beginning. In column (1) of Table 11 we see the coefficients for those factors. We see that *changingPrices* has a

Factors/Covariates	(1)	(2)	(3)
Intercept	0.639**** (0.0441)	0.277**** (0.0454)	0.252**** (0.0605)
numChoices=3	-0.178**** (0.0459)	-0.176**** (0.0438)	-0.175**** (0.0430)
numChoices=4	-0.106**** (0.0278)	-0.109**** (0.0276)	(0.0279)
numChoices=5	0.015 (0.0291)	0.015 (0.0292)	0.021 (0.0308)
numChoices=6	0	0	0
changingPrices=0	-0.169**** (0.0397)	-0.378**** (0.0176)	-0.378 (0.0169)
lambda		0.085**** (0.0084)	0.085**** (0.0081)
female=0			-0.005 (0.0160)
7-secondGame			-0.015 (0.0231)
gameCounter			0.001 (0.185)
Model Fit (QICC)	144.177	134.579	140.134

Table 11: GEE for the dependent variable *Realized Efficiency*. 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.

highly statistically significant effect on efficiency (as we expected). The coefficients for *numChoices* are with respect to the efficiency for *numChoices=6*. We see that the effect of *numChoices=3* and *numChoices=4* is statistically significant at $p < 0.001$. Furthermore, the coefficient for *numChoices=5* is positive, but it is not statistically significant. Thus, the efficiency does plateau at *numChoices=5*, but the data does not provide enough evidence that there is also a statistically significant *decrease* in efficiency as we go from 5 to 6 choices. In future studies we plan to conduct additional experiments with 7 or 8 choices, to find out if efficiency only plateaus, or eventually also decreases.

In column (2) of Table 11 we add the covariate lambda to the analysis. We see that *Lambda* has a statistically significant positive effect on efficiency, which makes sense because *Lambda* is a measure for the degree of rationality of each user. However, adding *Lambda* to the analysis does not result in any qualitative changes for the other results. Finally, in column (3), we add *Female*, *7-secondGame* and *GameCounter* to the analysis, only to show that they do not have a statically significant effect on efficiency. Note that we do not further investigate the design lever *Fixed vs. Changing Prices* at this point, because the optimal efficiency of the games with fixed and changing prices was very different, and thus doesn't allow for a meaningful comparison of the *Realized Efficiency*.[6]

[6] As mentioned before, our users also played a sequence of games with an overall time limit of 4 minutes where they had to trade-off spending more time on an individual decision with playing more game overall. In these games, we did find a statistically significant effect of the design lever *Fixed vs. Changing Prices* on the decision time. In particular, users needed more time to make a decision when prices where changing compared to when prices stayed

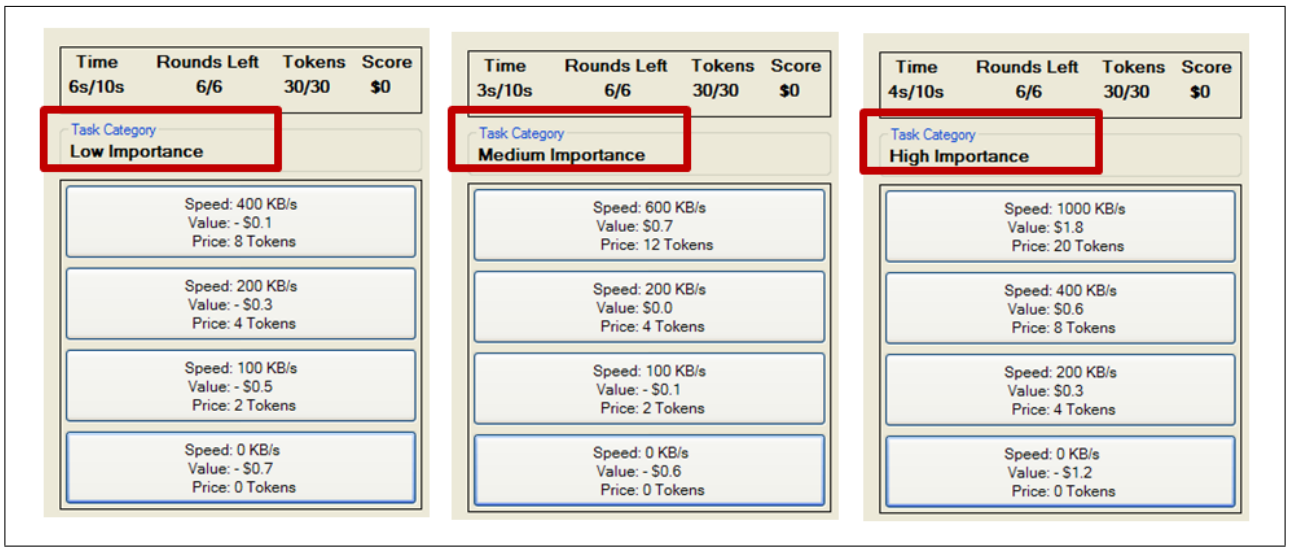


Figure 5: Adaptive Choice Sets: 3 different screenshots demonstrating the adaptive choice set idea. The users are offered a different set of choices (i.e., speed levels) depending on the current task category.

4.3.4 Results for Design Lever 3: 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 *UI Optimization*. 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. An intelligent agent can never truly know a user's current value for high bandwidth (or any other good/service for that matter); however, 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. For example, we could learn a mapping from context to a value estimate. Imagine that when a user is watching a streaming video or listening to Internet radio, he is 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, or reading an online newspaper. Over time, the application could learn this behavior, inferring that the user has a higher value for bandwidth when using the video or 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 better optimize his choices.

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 5 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 side, the choices are now better tailored to the individual decision situation. On the other side, the user now has to deal with the fact that the choices available to him (and thus

fixed. However, the analysis of this data is still underway and thus we are not presenting the detailed results in this thesis.

Factors/Covariates	(1)
Intercept	0.405**** (0.0410)
AdaptiveChoiceSets?	0.077** (0.0376)
Model Fit (QICC)	106.552

Table 12: GEE for dependent variable *RealizedEfficiency* 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.

also the prices) keep changing every round. The question is whether both effects taken together are positive or negative for the user's efficiency.

For design lever *Fixed vs. Adaptive Choice Sets* we also performed a statistical analysis for the dependent variable *OptChoice*. We found that having adaptive choices increased these users' likelihood of selecting the optimal action with high statistical significance (we omit the details for this particular analysis). Now, to see the effect of this design lever on efficiency, consider Table 12 where we show the results of fitting the generalized estimating equations to the data of study 2 (with the identity link function and assuming a normal distribution), where the dependent variable is *RealizedEfficiency*. We see that the coefficient for *AdaptiveChoiceSets?* is positive and statistically significant at $p < 0.05$. Thus, the data provides evidence that the introduction of adaptive choices indeed helped the users and resulted in significantly higher efficiency. This was not clear a priori, because having the composition of the choice set change in every round also makes the UI more complex and thus potentially increases the cognitive load on the users. However, apparently the negative effect of having more variability was significantly smaller than the positive effect of being able to make better decision, as the choices available are better tailored to the specific situations.

4.3.5 Results for Design Lever 4: UI Optimization

The fourth design lever we study is *UI Optimization*, where we optimize the market UI assuming 1) optimal play (i.e., modeling the user as being perfectly rational), or 2) suboptimal play (using a behavioral user model). For the UI assuming optimal play, we used the same algorithm as before, i.e., selecting the choice set composition with the highest optimal efficiency, i.e., where the corresponding MDP had the highest expected value. We used the experimental data obtained in the first study to find the best UI assuming sub-optimal play, taking into account the boundedly-rational behavior of real users. Figure 6 shows a diagram illustrating the "market UI optimization methodology" we employed.

The first step in Figure 6 corresponds to running study 1, where we obtained approximately 7,000 data points that we can use in our analysis, where each data point represents one action taken by a user in a particular game situation. The second and third step in the optimization method consists of learning a predictive user model, i.e., a model than is able to predict users' action choices in different game situations. For that user model, we use the quantal-response model described earlier. We computed different likelihood-maximizing λ-parameters 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

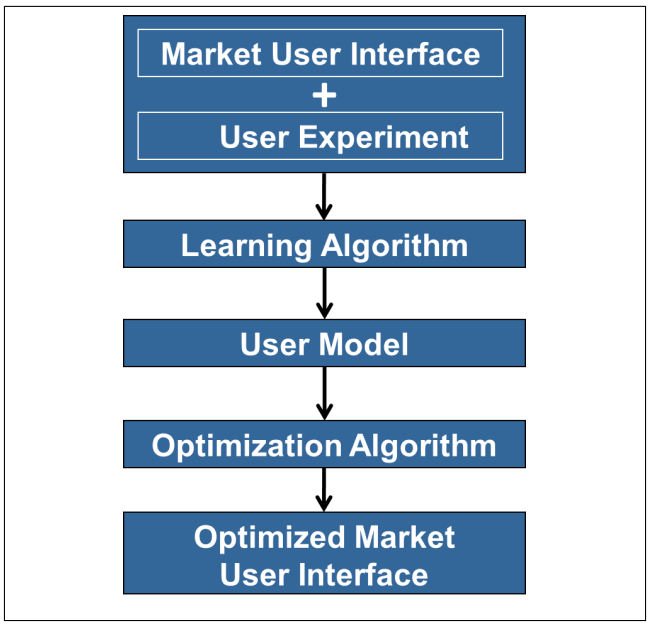
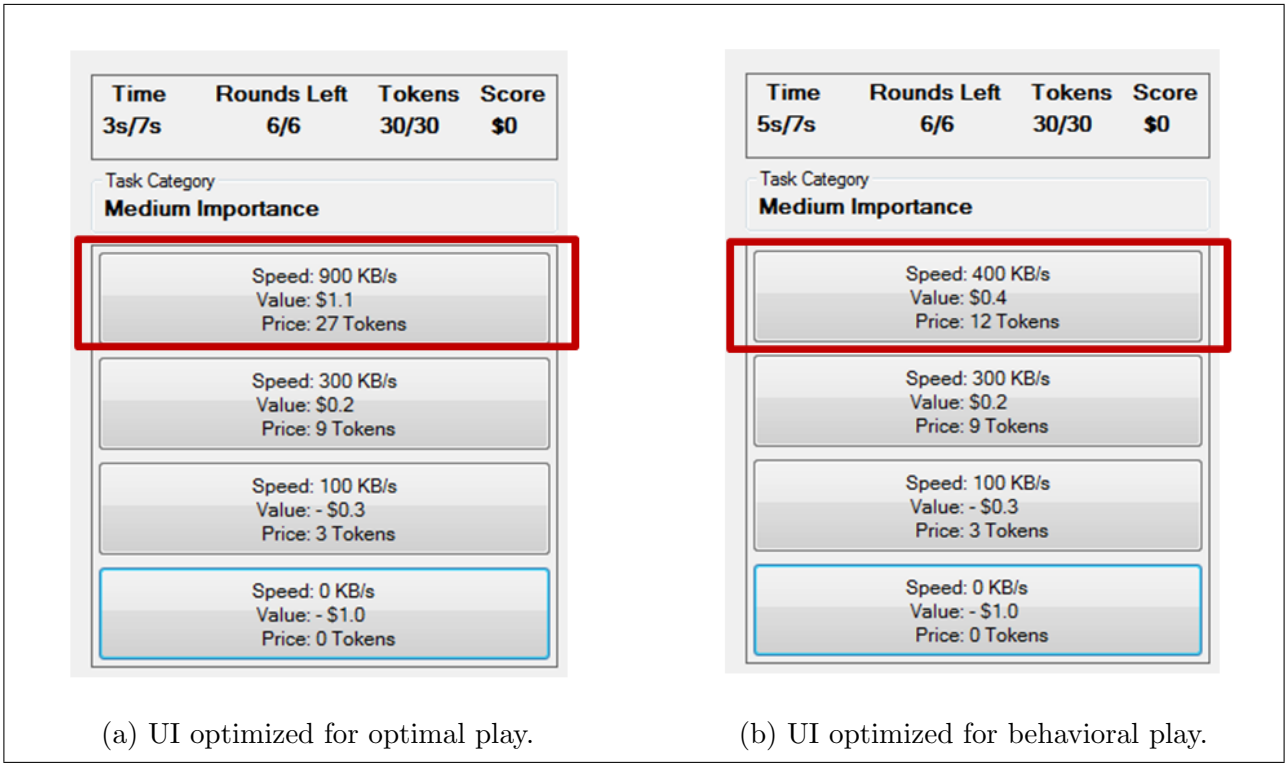


Figure 6: Market UI Optimization Method.

fixed or changing. Furthermore, we only considered the 7-second time treatment for the computation of λ because we expected to see the strongest differences in the 7-second games.

When studying design levers 1, 2, and 3, we solved the games optimally and computed the expected value of the game assuming perfect play. Equipped with the user model learned in step 3), we can now compute the expected value of a game assuming sub-optimal play by human players. For every configuration of the choice sets, this leads to a different expected value of the game. Thus, in step 4 of the optimization process depicted in Figure 6 we use the learned user model to search through the UI design space, i.e., through all possible configurations of choice sets and compute the expected value of the game according to the user model. We then choose the configuration with the highest expected value as the UI for "sub-optimal play". Consider Figure 7 where we display two screenshots for the games with fixed choice sets, illustrating the different choice sets resulting from the different UI optimization methods. In Figure 7(a), the UI optimized for optimal play is shown, and in Figure 7(b), the UI optimized for sub-optimal (behavioral) play is shown. Note that both UIs are *not* hand-picked, but the result of a computational search algorithm. We see that the only difference between the two UIs is the top choice: the UI that was optimized for optimal play gives the user the 900KB/s choice, while the UI that was optimized for sub-optimal play gives the user the 400KB/s choice. This result is understandable in light of how the UI-optimization algorithm works. The quantal-response assigns each action a certain likelihood of being chosen, corresponding to the Q-values of those actions. Now, consider the top choice in Figure 7(a), which has a high value, but which can also cost between 9 and 27 tokens (this is a game with changing prices). Thus, in the worst case, the user spends 27 out of his 30 tokens with one click, and then has only 3 tokens left for the remaining 5 rounds. Even if this action is very unlikely, the negative effect of an occasional mistake would be very large. Consequently, the UI optimized for sub-optimal play shown in Figure 7 does not have such high-value high-cost choices, reducing the negative effect of mistakes.

As before, we studied the effect of this design lever on *OptChoice* and found that the user's likelihood



(a) UI optimized for optimal play.

(b) UI optimized for behavioral play.

Figure 7: A screenshort of the game from Experiment 2, illustrating the differences in the user interface when (a) optimized for optimal play, and when (b) optimized for behavioral play.

of selecting the optimal choice increased. Thus, the optimization based on the behavioral model made the decision easier for the users. However, the efficiency results for this particular design lever are more complex and interesting.

Consider column (1) of Table 13 for the effect of design levers 3 and 4 on *Realized Efficiency*. We see that the coefficient for *OptimizedForSubOpt?* is *negative* and statistically significant at $p < 0.001$. The optimization of the market UI assuming behavioral play actually had a negative effect on *Realized Efficiency*. Thus, the behavioral model built around the quantal-response model and fitted to the data from Experiment 1 did *not* predict the users' decisions for study 2 accurately enough. Based on the behavioral model, the *Realized Efficiency*from the UI optimized for optimal play should have been significantly lower than the *Realized Efficiency*when playing the game with the UI optimized for sub-optimal UI. Section 4.2, which contains the analysis of the various behavioral factors on the user's decision making performance, offers a possible explanation for this effect. The UI optimization was based on the quantal-response model which only takes the Q-values of the different choices into account. It does not take into account 1) the number of choices available, 2) the relative or absolute position of the optimal choice, 3) whether the optimal choice had a positive or negative value, 4) the value of the choice one above the optimal choice, etc., even though we have found that all of these factors are highly statistically significant for the users' decision performance. Thus, one possible explanation is that the behavioral model we used was too simple, and didn't capture enough of the users' behavior to suffice for a good UI optimization.

Let's now take a more detailed look at the efficiency results. Table 14 provides at least a partial explanation for what happened. By re-optimizing the UI, we decrease the optimal efficiency (achievable

Factors/Covariates	(1)	(2)	(3)
Intercept	0.462**** (0.0501)	0.004 (0.0639)	0.053 (0.1417)
AdaptiveChoiceSets?	0.077** (0.0376)	0.08** (0.0367)	0.080** (0.0365)
OptimizedForSubOpt?	-0.111**** (0.0334)	-0.119**** (0.344)	
Lambda		0.103**** (0.0110)	0.100**** (0.0253)
SmallLambda=1			-0.065 (0.0530)
OptimizedForSubOpt *smallLambda=1			-0.069 (0.0500)
OptimizedForSubOpt *SmallLambda=0			-0.174**** (0.0391)
Model Fit (QICC)	106.927	98.265	101.895

Table 13: GEE for the dependent variable *RealizedEfficiency* studying the effect of *OptimizedForSubOpt*. 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.

for a perfectly rational player) from 1.0218 to 0.7819. Thus, we "took away" approximately \$0.24 per game. However, we never expected the users to come even close to the optimal efficiency values, but instead, based on our user model learned from study 1, we expected the users to do better in the re-optimized game such that the *Realized Efficiency*would actually increase. However, as we can see in the last column of Table 14, the *Realized Efficiency*also dropped from 0.42 to 0.3296. Thus, relative to the optimal efficiency, the users did better in the re-optimized game; however, in absolute terms, they still did worse. A potential explanation is that the users in study 2 acted "more rationally" than the users in study 1. However, the best fitting λ-parameters for study 1 and study 2 were very similar, and thus, the data does not support this hypothesis. Yet, we found another interesting result. As before in study 1, we computed a λ_i-parameter for each user in study 2, as well as one λ corresponding to the best fit across all users. In addition, we compute a binary variable *smallLambda* for each user which denotes whether that user's λ is smaller or larger than the *average* lambda, i.e., *SmallLambda* denotes whether the user belongs to the *more rational* or to the *less rational* group of users. Consider now column (3) of Table 13 where we also analyze the interaction effect of *OptimizedForSubOpt* and *SmallLambda*. We now see that for SmallLambda=0 (i.e., for the more rational users) the effect of *OptimizedForSubOpt* is particularly negative, i.e., for those users we made the game a lot worse by doing the re-optimization. However, for SmallLambda=1 (i.e., the less rational users) the effect of *OptimizedForSubOpt* is close to zero, and in fact not statistically significant. Thus, the data suggests that the less rational users did as well in the game whose UI was optimized for behavioral play as in the game whose UI was optimized for optimal play.

OptimizedForSubOpt?	*Optimal Efficiency*	*Realized Efficiency*
no	1.0218	0.4200
yes	0.7819	0.3296

Table 14: UI-Optimization: Effects on optimal and realized Efficiency.

5 Summary

In this paper, we have introduced a new research agenda on "market user interface design." Our goal is to understand how UI design choices for market environments affect users' abilities to make good economic decisions, and how we can develop automated methods to optimize market user interfaces. In studying this question, it is crucial to take the human nature of market participants into account, i.e., deviating from a perfectly ration agent model. Thus, our research explores a very complex space where human limited cognition meets computing. This is a largely unstudied research area with huge opportunities for work at the intersection of market design, intelligent agent systems, UI design, and behavioral economics. We situate our study in a 3G bandwidth market where users can make different choices regarding bandwidth speed on their smartphones for different prices. We designed a multi-step market game and ran a behavioral economics lab experiment with 53 users, testing the effect of four different design levers. The game can formally be modeled as an MDP and thus our work also provides insights into how well humans can play MDPs under time pressure. Our experimental results indicate that the users' actions were highly correlated with the Q-values of the choices available in the game, indicating that the users found very good sequential policies. In our analysis, we identified a series of behavioral effects. Perhaps one of the most important results concern the users' loss aversion without exhibiting any learning effects over time. This finding raises concerns about users' general ability to allocate a fixed budget over time in real-world domains.

Finally, we tested the effect of four different market UI design levers on users' *Realized Efficiency*. When changing the number of choices, the *Realized Efficiency*increases as we go from 3 to 4 to 5 choices, and then slightly decreases as we go from 5 to 6 choices. However, the decrease in efficiency was not statistically significant. Thus, it seems that after some point, adding more choices (thereby making the UI more complex), doesn't help the user, and can potentially even hurt. In future research, we want to study the effect of changing the number of choices in even more detail, running a similar experiment and adding 7 or 8 choices to the treatments to see if efficiency merely plateaus at some point, or even starts to decrease again. In a second experiment, we studied the effect of the two design levers *Fixed vs. Adaptive Choices* and *UI Optimization*. Our results show that having adaptive rather than fixed choice sets has a positive effect on users' Realized Efficiency. This is a positive result, suggesting numerous applications where user interfaces could be tailored in various ways to context-specific needs of the users.

In contrast, and quite surprisingly, we found that the UI optimization using the quantal-response model was not successful. In fact, the UI that was optimized based on the behavioral model actually led to lower *Realized Efficiency* than the UI optimized for optimal play. However, more interestingly, we found a very large, statistically significant difference between the *less rational* and the *more rational* users. For the more rational users, the UI re-optimization led to a significantly lower efficiency, while there was no statistically significant effect on efficiency for the less rational users. This finding naturally suggests a new research direction on "personalized market user interfaces." In many domains, in particular in the smartphone domain, there is a lot of user-specific, behavioral and non-behavioral data available that carries a lot of information about the particular user. If we can estimate a user's

"degree of rationality" based on this data, we can provide each user with a market UI that is specifically optimized for that particular (kind of) user. Taking this idea a step further, we can also estimate a user's value for time and take this into account in the UI personalization. Thus, there are still many opportunities in this space, ranging from more complete behavioral models to algorithms for learning user preferences and automated UI optimization.

References

Georg Buscher, Susan Dumais, and Edward Cutrell. 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, July/August 2010.

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

James Choi, David Laibson, and Brigitte C. Madrian. Plan Design and 401(k) Savings Outcomes. *National Tax Journal*, 57(2):275–298, 2004.

Mary Czerwinski, Desney S. Tan, and George G. Robertson. Women Take a Wider View. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, Minneapolis, MN, April 2002S.

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

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

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

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

Sheena S. Iyengar and Wei Jiang. Choosing Not to Choose: The Effect of More Choices on Retirement Savings Decisions. Columbia University Working Paper, 2003.

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

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

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

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

John W. Payne and James R. Bettman. Preferential Choice and Adaptive Strategy Use. In Gerg Gigerenzer and Reinhard Selten, editors, *Bounded Rationality: The Adaptive Toolbox*. MIT University Press, Cambridge, MA, 2001.

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

Todd Sarver. Anticipating Regret: Why Fewer Options May be Better. *Econometrica*, 76(2):263–305, 2008.

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

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

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

Richard H. Thaler, Cass R. Sunstein, and John P. Balz. Choice Architecture. Unpublished. Available at SSRN: http://ssrn.com/abstract=1583509, April 2010.

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