Interface Design Optimization as a Multi-Armed Bandit Problem

J. Derek Lomas[1], Jodi Forlizzi[2], Nikhil Poonwala[2], Nirmal Patel[2], Sharan Shodhan[2], Kishan Patel[2], Ken Koedinger[2], Emma Brunskill[2]

The Design Lab[1]
UC San Diego
9500 Gilman Drive
dereklomas@gmail.com

Carnegie Mellon University[2]
HCI Institute,
5000 Forbes Ave
{forlizzi,krk,ebrun}@cs.cmu.edu

ABSTRACT

"Multi-armed bandits" offer a new paradigm for the AI-assisted design of user interfaces. To help designers understand the potential, we present the results of two experimental comparisons between bandit algorithms and random assignment. Our studies are intended to show designers how bandits algorithms are able to rapidly explore an experimental design space and automatically select the optimal design configuration. Our present focus is on the optimization of a game design space.

The results of our experiments show that bandits can make data-driven design more efficient and accessible to interface designers, but that human participation is essential to ensure that AI systems optimize for the right metric. Based on our results, we introduce several design lessons that help keep human design judgment in the loop. We also consider the future of human-technology teamwork in AI-assisted design and scientific inquiry. Finally, as bandits deploy fewer low-performing conditions than typical experiments, we discuss ethical implications for bandits in large-scale experiments in education.

Author Keywords

Educational games; optimization; data-driven design; multi-armed bandits; continuous improvement;

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

The purpose of this paper is to provide empirical evidence that can help designers understand the promise and perils of using multi-armed bandits for interface optimization. Unlike the days of boxed software, online software makes it easy to update designs and measure user behavior. These properties have resulted in the widespread use of online design experiments to optimize UX design. On any given day, companies run thousands of these online controlled experiments to evaluate the efficacy of different designs [21]. These experiments are changing the nature of design by introducing quantitative evidence that can serve as a "ground truth" for design decisions. This evidence, however, often conflicts with designer expectations. For instance, at Netflix and Google [22,33] only about 10% of tested design improvements (which were, presumably, introduced as superior designs) actually lead to better outcomes.

What is "good design" is necessarily relative. However, a "better" design is often defined as a design with better outcome metrics. Yet, the relationship between metrics and "good design" is not always clear. Design experiments seemingly show that particular designs are *objectively* better. However, these metrics are not always conclusive measures of value, though they can appear to be. Even when a particular design gets better results in A/B test, it is not always the best design for the larger objectives of the organization [9,22].

This paper addresses these issues through a set of experiments involving different design optimization algorithms. We present empirical data showing how bandit algorithms can increase the efficiency of experiments, lower the costs of data-driven continuous improvement and improve overall utility for subjects participating in online experiments. However, our data also show some of the limitations of relying on AI without human oversight. Together, our evidence can help designers understand why design optimization can be seen as a "multi-armed bandit" problem and how bandit algorithms can be used to optimize user experiences.

RELATED WORK

The optimization of interfaces based on individual user characteristics is a significant sub-field within HCI. One challenge in the field has been the availability of these evaluation functions at scale. For instance, SUPPLE [12], a system for algorithmically rendering interfaces, initially required cost/utility weights that were generated manually for each interface factor. Later, explicit user preferences

CHI'16, May 7-12, 2016, San Jose, CA, USA

ACM 978-1-4503-3362-7/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858425

were used to generate these cost/utility weights. Other algorithmic interface optimization approaches have used cost functions based on estimated cognitive difficulty, user reported undesirability, user satisfaction ratings and even aesthetic principles [11, 42].

Online user data and A/B testing provide an alternative source for UX evaluation. Hacker [16] contrasts UX optimization for individual users (*student adaptive*) with the optimization of interfaces for all users (*system adaptive)*. In Duolingo (a language learning app used by millions), student adaptive UX is supported by knowledge models, whereas Duolingo's system adaptivity is a longer term process that unfolds in response to large-scale A/B tests. A/B testing is also widely used by Google, Amazon, Yahoo, Facebook, Microsoft and Zynga to make data-driven UX decisions on topics including surface aesthetics, game balancing, search algorithms and display advertising [22]. With that framing, A/B testing is indeed a widespread approach for the optimization of interface design.

Design Space Optimization

Design space optimization involves finding the set of design factor parameter values that will maximize a desired outcome. This follows Herb Simon's [38] notion that designing can be understood in terms of generating design alternatives and searching among those alternatives for designs that best satisfy intended outcomes. The concept of "design spaces" has been used to formally represent design alternatives as a multiplication of independent design factors [7]; for instance, the color, size and font of text are independent design factors. While the notion of design space exploration as a mechanism for optimization is commonly used in areas like microprocessor design, there is also a rich history of the concept in HCI, where design spaces have been used to express designs as a possibility space, rather than as a single final product [33].

In this paper, we focus on the optimization of a game design space, or the multiplication of all game design factors within a particular game [30]. The complete design space of a game (all possible variations) can be distinguished from the far smaller *instantiated design space*, which consists of the game variables that are actually instantiated in the game software at a given time. Games are a rich area for design space exploration as they often have many small variations to support diverse game level design. During game balancing, designers make many small modifications to different design parameters or design factors. Thus, the instantiated design space of a game will often include continuous variables like reward or punishment variables (e.g., the amount of points per successful action or the effect of an enemy's weapon on a player's health) and categorical game design variables (e.g., different visual backgrounds, enemies or maps). Increasingly, online games use data to drive the balancing or optimization of these variables [21].

Even simple games have a massive design space: every design factor produces exponential growth in the total number of variations ("factorial explosion"). For instance, previous experiments with the game presented in this paper varied 14 different design factors and dozens of factor levels [29]; in total, the instantiated design space included 1.2 billion variations! Thankfully, design isn't about completely testing a design space to discover the optimum, but rather to find the most satisfactory design alternative in the available time. Designers often use their imaginative judgment to evaluate design alternatives, but this is error prone [22]. Thus, this paper explores multi-armed bandit algorithms as a mechanism for exploring and optimizing large design spaces.

Multi-Armed Bandits and Design

The Multi-Armed Bandit problem [25] refers to the theoretical problem of how to maximize one's winnings on a row of slot machines, when each machine has a different and unknown rate of payout. The solution to a multi-armed bandit problem is a selection strategy; specifically, a policy for which machine's arm to pull to maximize winnings, given the prior history of pulls and payoffs. The objective is normally to minimize *regret*, which is often defined as the difference between the payoff of one's policy and the payoff of selecting only the best machine – which, of course, is unknown a priori.

This paper frames UX optimization design decision-making as a multi-armed bandit problem [25]: Each variation of a UX design can be viewed as a slot machine "arm" where the payoff (in our case, user engagement) is unknown. We view data-driven designers as needing to balance exploring designs that are potentially effective with the exploitation of designs that are known to be effective. This framing allows us to draw upon the extensive machine learning research on algorithms for solving multi-armed bandits problems.

Solving Multi-Armed Bandit Problems

If one doesn't know which slot machine has the highest payout rate, what can be done? For instance, one could adopt a policy of "explore first then exploit". In this case, one could test each machine *n* number of times until the average payout of a particular arm is significantly higher than the rest of the arms; then this arm could be pulled indefinitely. This "explore first then exploit" approach is similar to traditional A/B testing, where subjects are randomly assigned to different conditions for a while and then one design configuration is chosen as the optimal. However, this approach is often "grossly inefficient" (p. 646, [36]): for instance, imagine the extreme example where one arm produces a reward with every pull while all other arms don't pay out at all. In a typical experiment, the worst arm will be pulled an equal amount of time as the most optimal arm. Furthermore, the stopping conditions for A/B tests are quite unclear, as few online experimenters properly determine power in advance [21]. As a result, this runs the risk of choosing the (wrong) condition too quickly.

A Real-World Illustration

Consider a hypothetical game designer who wants to maximize player engagement by sending new players to the best design variation: A, B or C (Figure 1). How should she allocate the next 1000 players to the game? She could pick the single condition that has the highest average engagement at that time. Alternatively, she could have a policy of sending players to the design condition with the highest mean ("C"). However, this "greedy" approach doesn't take into account that A's mean might be higher than C, as there isn't enough information yet.

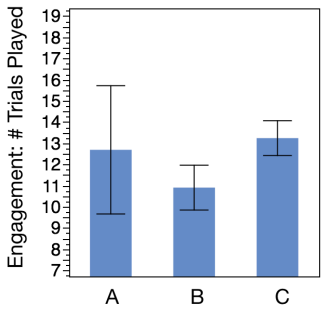

Figure 1: Example data comparing three different design conditions. While "C" has the highest mean, "A" has the highest upper confidence limit. The policy of picking the condition with the highest upper confidence limit can improve returns by balancing exploration with exploitation.

So instead, our designer could have a simple policy of just choosing the design that has the highest upper confidence limit. This policy would result in testing the engagement of A with additional players. The additional information gathered from A would either show that condition A actually does have a higher mean than condition C, or it would shrink the confidence interval to the point where the policy would begin to pick C again.

Practically, this policy of picking the higher upper confidence limit has the effect of choosing designs that have either a higher mean (exploitation) or insufficient information (exploration). This policy was instantiated as the UCL-bandit in order to help communicate the multi-armed bandit problem to a general audience. It is not expected to be a practical algorithm for solving real-world bandit problems.

Background on Multi-Armed Bandits

In the past few years, multi-armed bandits have become widely used in industry for optimization, particularly in advertising and interface optimization [22]. Google, for instance, uses Thompson Sampling in their online analytics platform [37]. Yahoo [26] uses online "contextual bandits" to make personalized news article recommendations. Work has also been done to apply bandits to basic research, as well as applied optimization. Liu *et al.* [27], for instance, explored the offline use of bandits in the context of online educational games, introducing the UCB-EXPLORE algorithm to balance scientific knowledge discovery with user benefits.

There are many multi-armed bandits algorithms that sequentially use the previous arm pulls and observations to guide future selection in a way that balances exploration and exploitation. Here we present a non-comprehensive review of approaches to the bandit problem in the context of online interface optimization.

Gittins Index: This is the theoretically optimal strategy for solving a multi-armed bandit problem. However, it is computationally complex and hard to implement in practice. [14,6]

Epsilon-First: This is the classic A/B test. Users are randomly assigned to design conditions for a set time period, until a certain number of users are assigned, or until the confidence interval between conditions reaches a value acceptable to the experimenters (e.g., test until $p<0.05$). After this time, a winner is picked and then used forever.

Epsilon-Greedy: Here, the policy is to randomly assign x% of incoming users, with the remainder assigned to the highest performing design. A variation known as *Epsilon-Decreasing* gradually reduces x over time; in this way, exploration is emphasized at first, then exploitation [2]. These algorithms are simple but generally perform worse than other bandit techniques.

Probability Matching: Players are assigned to a particular condition with a probability proportionate to the probability that the condition is optimal [36]. Thus, some degree of random assignment is still involved. Probability matching techniques include Thompson Sampling [36,37,38] and other Bayesian sampling procedures that are used in adaptive clinical trials [3].

UCB1: After testing each arm once, users are assigned to the condition with the highest upper confidence bound. With slight abuse of notation, in this paper we will use the word "limit" to imply that the data is assumed to be generated from a normally distributed variable, and "bound" to make no assumption on the underlying data distribution process (except that there exists bounds on the possible data values) UCB was chosen for the present study due to its strong theoretical guarantees, simplicity and computationally efficiency.

UCL: In this paper, we introduce an illustrative approach to bandit problems using upper confidence limits for assignment, which is intended to help those familiar with basic statistics to understand the logic underlying the UCB algorithm. Again, here we slightly abuse the term "limit" to compute confidence intervals that assume the data is normally distributed. The benefit of doing this is that the confidence intervals are typically much tighter than if one makes no such assumption (as in UCB). This algorithm operates by calculating the upper confidence limit of a design condition after first randomly assigning a condition for a period of time (e.g., 25 game sessions); every new player thereafter is assigned to the condition with the highest upper confidence limit.

Recall than an X% confidence interval [a,b] around a parameter of interest is computed from data such that if the experiment was repeated many times, at least X% of the time the true parameter value would lie in the computed interval for that experiment. For example, this parameter could be the mean outcome for a design condition. The confidence limits are precisely the two edges of the interval, a and b. Confidence intervals can be computed under different assumptions of how the data is generated given the parameter of interest (e.g. sampled from a normal distribution with a particular mean, etc.).

System for Bandit-Based Optimization

We present a system for integrating bandit-based optimization into a software design process with the following goals: 1) Support the use of data in an ongoing design process by communicating useful information to game designers; 2) Automatically optimize the design space identified by designers in order to reduce the cost of experimentation and analysis. 3.) Reduce the cost of experimentation to the player community by minimizing exposure to low-value game design configurations. Our data-driven UX optimization method extends previous optimization research involving user preferences [12], aesthetic principles [11], and embedded assessments [13]. Moreover, we extend previous applications of bandits for offline educational optimization [27] by demonstrating the value of bandits as a tool for online optimization.

Measures

Optimization relies on an outcome evaluation metric [12] that drives the decisions made between different design configurations. Choosing an appropriate outcome variable is essential, because this metric determines which conditions are promoted. In this paper, our key outcome metric for engagement is the number of trials (estimates) that are made, on average, within each design variation.

EXPERIMENT 1

Our first "meta-experiment" compares three different algorithms for conducting online experimentation (we use the term "meta-experiment" because it is experimenting with different approaches to experimentation). The players of an educational game were first randomly assigned to one of these experiments. Then, according to their assignment method, the players were assigned to a game design within the experimental design space of *Battleship Numberline.*

We hypothesize that, in comparison to random assignment, one of the multi-armed bandit assignment approaches will result in greater overall student engagement during the course of the study. Specifically, we hypothesize that multi-armed bandits can automatically optimize a UX design space more efficiently than random assignment and also produce greater benefits and lower costs for subjects. Costs occur when participants receive sub-optimal designs.

H1: Multi-Armed Bandits will automatically search through a design space to find the best designs

H2: Automatic optimization algorithms will reduce the cost of experimentation to players

To test our hypothesis, we simultaneously deployed three experiments involving 1) random assignment 2) UCB1 bandit assignment or 3) UCL-95% bandit assignment. The UCL-95% bandit had a randomization period of 25, meaning that all design variations needed to have 25 data points before it started assignment on the basis of upper confidence interval.

Calculating UCB1: In this paper, we calculate the upper confidence bound of a design "D" as the adjusted mean of D (the value must be between 0-1, so for us, we divided each mean by the maximum engagement allowed, 100) + square root of (2 x log(n) / n_D), where n is the total number of games played and n_D is the total number of games played in design condition D.

Calculating UCL: The upper confidence limit is calculated as the mean + standard error x 1.96 (for 95% confidence) and mean + standard error x 3.3 (for 99.9% confidence).

The Design Space of Battleship Numberline

Battleship Numberline is an online game about estimating fractions, decimals and whole numbers on a number line. Players attempt to blow up different targets by estimating the location of different numbers on a number line. The "ship" variant (in the xVessel factor) requires users to type a number to estimate the location of a visible ship on a line. The "sub" variant requires users to click on the number line to estimate the location of a target number indicating the location of a hidden submarine (the sub is only shown after the player estimates, as a form of grounded feedback [40]).

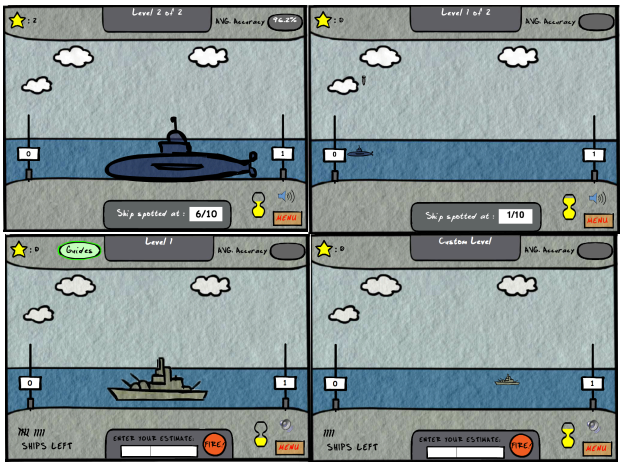

Figure 2: Variations of the xVessel and the xPerfectHitPercent design factors. xVessel variants are submarines (hidden until a player clicks on the line to estimate a number) and ships (players type in a number to estimate where it is on the line). xPerfectHitPercent varies the size of the target, in term of the accuracy required to hit it. The above targets require accuracies of 70% (largest), 95%, 80% and 97% (smallest) .

The size of the targets (xPerfectHitPercent) represents the level of accuracy required for a successful hit. For example,

when 95% accuracy is required, the target is 10% of the length of the number line—when 90% accuracy is required, the target is 20% of the line. Thus, big targets are easier to hit and small targets are more difficult.

The combination of these, and other, design factors constitute the design space of *Battleship Numberline*. The present study evaluates different methods for exploring and then selecting optimal design condition within this design space. In this paper, we only explore the xVessel and xPerfectHitPercent design factors.

Procedure

Over the course of 10 days, 10,832 players of *Battleship Numberline* on Brainpop.com were randomly assigned to three different experimentation algorithms: random, upper confidence bound (UCB), and upper confidence limit (UCL). The UCL-95% bandit had a randomization period of 25 trials before assigning by UCL. Within each experimentation condition, players were assigned to 1 of 6 different conditions, a 2x3 factorial involving the xVessel (clicking vs. typing) and the size of targets (xErrorTolerance, where larger is easier to hit).

After players made a choice to play in the domain of whole numbers, fractions or decimals, they were given a four-item game-based pretest [29]. They were then randomly assigned to an experimental method, which assigned them to one of 12 experimental conditions. If players clicked on the menu and played again, they were reassigned to a new experiment.

When players enter the game, the server receives a request. It then allocates conditions to these incoming requests based on a round-robin method of assignment, that starts at the experimental level and then at the condition level. For instance, the player would be assigned to random assignment and then assigned to one of the random assignment conditions; the next player in the queue would be assigned to UCB. The UCB algorithm would then determine which game condition they would receive.

Experimental System

Our experimental system was built on Heroku and MongoDb. It was conceived with the intention of providing data analytics dashboard for designers to monitor the experiment. The dashboard shows running totals of the number of times each arm is pulled (i.e., the number of times a design condition was served to a player), the mean number of trials played in each condition (our measure of engagement), the upper confidence limit and the upper confidence bound of each condition. Our dashboard also provided a mechanism for designers to click on a URL to experience for themselves any particular design (i.e., any arms in the experiment). These links made it easy to sample the design space and gain human insight into the data. We also provided a checkbox so that a particular arm could be disabled, if necessary. Finally, our system made it easy to set the randomization period and confidence limit for the UCL algorithm.

RESULTS

In Figure 3 and Table 1, we confirm **H2** by showing that players in the bandit conditions were more engaged; thus, bandits reduced the cost of experimentation to subjects by deploying fewer low-engagement conditions.

The UCL and the UCB bandit experiment produced 52% and 24% greater engagement than the experiment involving random assignment. Our measure of regret between experiments is shown in Table 1 as the percent difference in our outcome metric between the different experiments and the optimal policy (e.g., Sub90, which had the highest average engagement). This shows that the UCL-25 experiment (one of the bandit algorithms) achieved the lowest regret of all experiments.

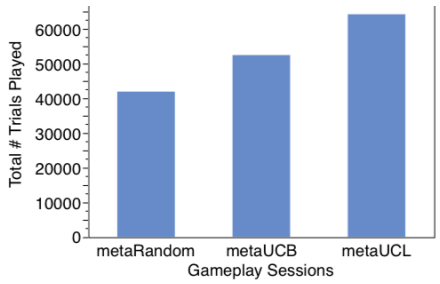

Figure 3: Shows how the bandit-based experiments garnered more student engagement (total number of trials played).

Meta Experimental Conditions	Total Games Logged	Total Trials Played	% Difference from Optimal
Random Assignment	2818	42,835	36%
UCB Assignment	2896	53,274	20%
UCL-25 Assignment	2931	65,206	2%
Optimal Policy (Sub90)	2931*	66,534*	0%

Table 1: Comparison of Meta-Experiment. * 22.7 average trials for Sub90. Assuming same number of games logged. 8,645 total logged out of 10,832 total served.

H1, the hypothesis that bandits can automatically optimize a UX design space, was confirmed with evidence presented in Figure 4. These data show that random assignment equally allocated all 6 conditions whereas both the UCB bandit and the UCL bandit allocated subjects to conditions preferentially. The reason for this unequal allocation is the difference in the efficacy of the conditions for producing player engagement, as seen in Figure 5.

Figure 5 shows the means and 95% confidence intervals of each arm in the three experiments. The Y-axis is our measure of engagement: the average number of trials played in the game within that condition. All experiments identify Sub90 as the most engaging condition. In this variant of the game, players needed to click to estimate a

number on the number line and their estimates needed to be 90% accurate to hit the submarine. All bandit-based experiments deployed this condition far more than other conditions (as seen in Figure 4).

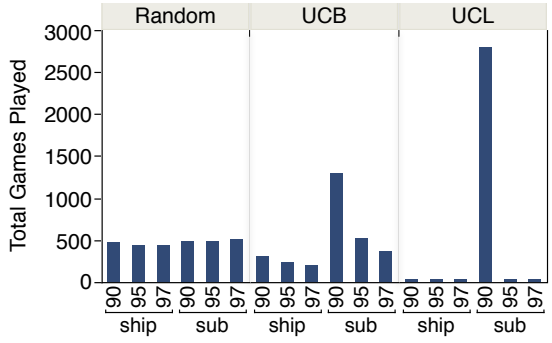

Figure 4: Random assignment experimentation equally allocates subjects whereas both bandit-based experiments allocate subjects based on the measured efficacy of the conditions. Total Games Played is the number of players assigned to each design condition.

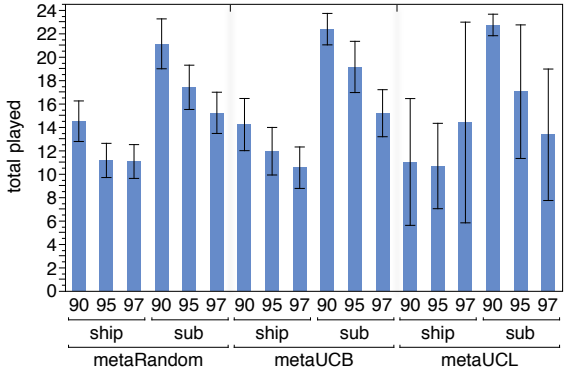

Figure 5: The means and 95% confidence intervals of the experimental conditions. Note that the Upper Confidence Limit experiment focused data collection on the condition with the highest upper confidence limit. Y-Axis represents Engagement, as the total number of trials played.

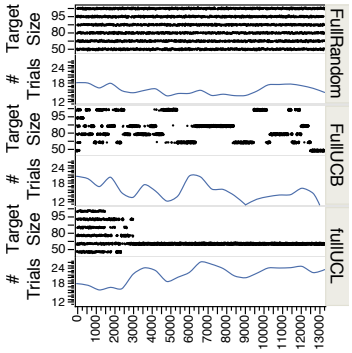

Figure 6: This graph shows the allocation of game sessions to the different target sizes and the variation in the average number of trials played over time.

Note the long confidence intervals on the right side of Figure 5: these are a result of insufficient data. As can be seen in Figure 4, these conditions each had less than 30 data points. However, if any of the confidence intervals were to exceed the height of Sub90's condition in the UCL experiment, then UCL would deploy those again.

EXPERIMENT TWO

After running the first meta-experiment, our results clearly supported the value of bandits. However, UCL never tested additional design variations after "deciding" during the randomization period that Sub90 was the most engaging condition (Figure 6). While the benefits of the Sub90 outcome was confirmed by the random experiment, it does not illustrate the explore-exploit dynamic of a bandit algorithm. Therefore, to introduce greater variation and bolster our discussion, we ran a second meta-experiment.

In this meta-experiment, we compared random assignment with two variations on the Upper Confidence Limit bandit. We retained our use of the greedy 95% confidence limit bandit but also added a more conservative 99.9% confidence limit bandit. The difference between these two is the parameter that is multiplied by the standard error and added to the mean: for 95% we multiply 1.96 times the standard error while for 99.9% we multiply 3.3. We expect this more conservative and less greedy version of UCL to be more effective because it is less likely to get stuck in a local optimum.

H3: UCL-99.9% will tend to deploy a more optimal design condition than UCL-95% as a result of greater exploration.

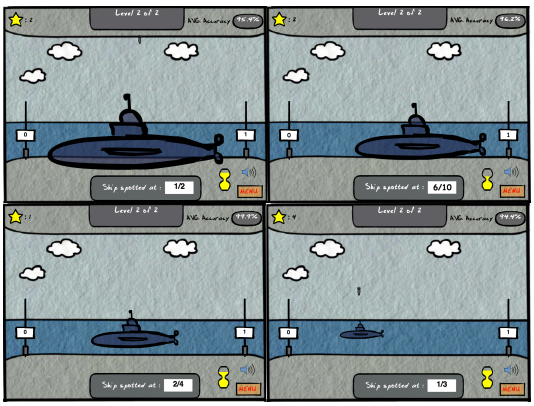

Figure 7: Four of the five variations in target size in experiment 2. The largest submarines (at top, 60 and 70) appeared to be far too big and too easy to hit. However, they were generally more engaging than the smaller sized targets.

In the second experiment, we focused on submarines, which we found to be more engaging than ships (or, in any case, resulted in more trials played). We eliminated the smallest size (which was the worst performing) and added a broader sample: 95%, 90%, 80%, 70%, 60% accuracy required for a hit. The largest of these, requiring only 60% accuracy for a hit, was actually 80% of the length of the number line. Although we felt the targets were grotesquely large, they actually represented a good scenario for using online bandits for scientific research. We'd like to collect online data to understand the optimal size, but we'd want to

minimize how many students were exposed to the suboptimal conditions. We were sure that this target size was too large, but could the bandits detect this?

H4: Bandits will deploy "bad" conditions less often than random assignment.

RESULTS

We found mixed evidence for both **H3** and **H4**. The more conservative bandit did not appear to explore longer (Figure 10) nor did it deploy better designs (Figure 8), although it did achieve greater engagement than UCL-95 (Table 2). Additionally, while the bandits did achieve less regret than the random assignment condition (Table 2), the conditions deployed were so bad that we got phone calls from Brainpop inquiring whether there was a bug in our game!

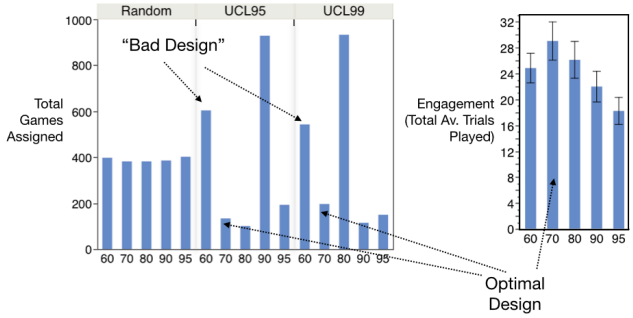

Figure 8: The "bad design" (Sub60) did surprisingly well in the bandits, where it was assigned second most often – and far less than the optimal design, according to random assignment.

Meta Experimental Conditions	Total Games Logged	Total Trials Played	% Diff. from Optimal
Random Assignment	1,950	46,796	21%
UCL-95%	1,961	47,312	20%
UCL-99.9%	1,938	49,836	14%
Optimal Policy (Sub70)	1,950*	56,628*	0%

Table 2: Compares each experiments to the optimal policy: Sub70 according to the random assignment experiment. * 29.04 average trials for sub70 in the random condition.

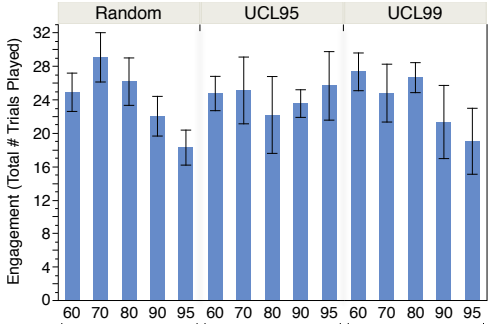

Figure 9: The means and confidence intervals of each condition within each experiment. The random assignment experiment reveals an inverted U-shape, where the largest target is less engaging than the second largest. The rank order of condition engagement varies over the experiments.

In Figure 8, we show that the *least* chosen arm of UC-95% was the *most* chosen arm of UCL-99.9% -- and vice versa! Moreover, the second most-picked design of both bandits was the very largest target, which seemed to us to be far too large. Finally, the optimal arm, according to random assignment, was hardly in the running inside the bandit experiments. What accounts for this finding?

First, all of the design variations in this experiment performed reasonably well (Figure 9) on our engagement metric, even the "bad" conditions. Without major differences between the conditions, all experimental methods performed relatively well. But, digging in, there were other problems that have to do with the dangers of sequential experimentation.

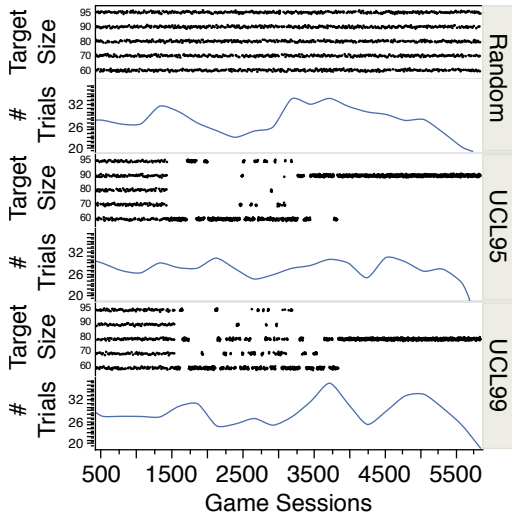

Figure 10: For meta-experiment 2, this graph shows the allocation of game sessions to the different target sizes and the variation in the average number of trials played over time.

Figure 10 shows the allocation of designs over time. The X-axis represents game sessions over time (i.e.,"arm pulls", as each game session requests a design from the server). The smoother line shows the mean number of trials played over time (i.e., engagement over time). Note that the randomly assigned mean varies significantly over time, by nearly 50%! This reflects the "seasonality" of the data; for instance, the dip in engagement between 2000 to 3000 represents data collected after school ended, into the evening. And the rise around 3000 represents a shift to the morning school day.

Therefore, the average player engagement in the bandits can be expected to be affected by the same seasonal factors as those affecting random assignment, yet also vary for other reasons. In particular, the average engagement of these bandits will be affected by the different conditions they are deploying. Possible evidence of this phenomenon can be noted in the concurrent dip in engagement around 4200 – a dip that is not present in random assignment. This dip occurs during the morning hours, so one plausible

explanation is that the morning population is not as engaged by the designs deployed by the bandits as much as other populations. It appears that students in the morning are more engaged by challenge—possibly because students in the morning have higher ability, on average.

In both meta-experiments, all bandit algorithms tended to test different design conditions in bursts. Note for instance that the Sub70, the optimal condition, was explored briefly by UCL-95% around 2500-3000 – yet this period was the time when all conditions were fairing the worst. So, this condition had the bad luck of being sampled at a time when any condition would have a low mean. This shows the limitations of sequential experimentation in the context of time-based variations.

We cannot confirm our hypothesis that the UCL-99.9% bandit actually explored more than UCL-95%. Visually, it appears that UCL-99.9% was busy testing more conditions more often than UCL-95%. However, both bandits fell into a local optimum at roughly the same time, based on when each began exclusively deploying one condition.

GENERAL DISCUSSION

Our studies explore the optimization of the design space of an educational game using three different multi-armed bandit algorithms. In support of using bandit algorithms over standard random experimental design, we consistently found that bandits achieved higher student engagement during the course of the experiment. While we focused on game design factors for optimization, this work is relevant to any UI optimization that seeks to manipulate independent variables to maximize dependent variables.

Our work is notable, in part, for the problems it uncovered with automated data-driven optimization. For instance, we were surprised to find that one of the most engaging conditions was Sub60 (the absurdly large condition in Figure 7), despite the fact that it was included for the purpose of testing the ability of the bandits to identify it as a poor condition. This discrepancy indicates that our metric (number of trials played) may be the wrong metric to optimize. Alternatively, the metric might be appropriate, but we (and Brainpop) might be wrong about what is best for users. Our work illustrates how automated systems have the potential to optimize for the wrong metric. The risks of AI optimizing arbitrary goals has also been raised by AI theorists [5]; one thought experiment describes the dangers of an AI seeking to maximize paperclip production.

Dangers of sequential experimentation in the real world

Our results also point to practical issues that must be understood and resolved for bandit algorithms to transition from computer science journals to the real world. For instance, we found that time-based variations (e.g., average player engagement was less during the night than during the day) significantly affected the validity of our sequentially experimenting bandits. These fluctuations due to contextual time-of-day factors have a much bigger effect on sequentially experimenting bandits than random assignment. So, even though much more data was collected about particular arms, it was not necessarily more trustworthy than the data collected by random assignment.

While it is likely that conservative UCB-style bandits would eventually identify the highest performing arm if they were run forever, these time-based variation effects can significantly reduce their performance. In contrast, these effects may help explain the remarkable success of simple bandit algorithms like *Epsilon-Greedy,* which randomize a portion of the traffic and direct another portion to the condition with the highest mean. Thompson Sampling also randomly assigns players to all conditions, but with lower probabilities for lower performing conditions. While known factors (like time-of-day) can be factored into bandit algorithms [26], any bandit that involves randomization (like Thompson Sampling) is likely to be more trustworthy in messy real-world scenarios.

Limitations

Our goal was to run an empirical study to illustrate how bandits work, not to introduce better algorithms. Our work might be viewed as specific to games; however, we view it in the context of any situation where experimentation with design factors (independent variables) might optimize outcome measures (dependent variables).

Our algorithms, however, were simple. For instance, they couldn't do things like take into account factors like time of day (like contextual bandits [26]). Additionally, our bandit algorithms did not allow for the sharing of information between arms or make use of previously collected data, which would be especially useful for continuous game design factors (such as the size of our targets) or larger design spaces. To this end, we might have considered approaches making use of fractional factorial designs, linear regressions or a "multi-armed mafia" [36]. Recent work combining Thompson Sampling and Gaussian Processes is also promising [18].

Our UCL bandit was designed to help explain how bandits work to a general audience, specifically, by illustrating the conceptual relationship between the mechanism of the UCB algorithm and the policy of "always choosing the design with the highest error bar (upper confidence limit)." In our experience, general audiences quickly grasp this idea as an approach for balancing exploration and exploitation. In contrast, far less familiarity with Chernoff-Hoeffding bounds (the basis for the UCB bandit). This illustrative value of the UCL algorithm is important because our goal is to contribute an understanding of UI design optimization as a multi-armed bandit problem to designers, not contribute a new bandit algorithm to the machine learning community.

Indeed, there are fundamental problems that should be expected from the UCL bandit. Constructing bandits that operate using confidence intervals is conceptually similar to running experiments and constantly testing for significance

and then running the condition that is, at the time, significant. However, significance tests assume that sample size was estimated in advance. While our bandits were "riding the wave of significance" they were susceptible to the tendency to be over confident in the present significance of a particular condition. This is a major problem in contemporary A/B testing, as well [11].

There are other fundamental issues with UCL. For instance, unlike UCB1, both UCL bandits had a tendency to fall into a local maximum for a long period of time, without exploration. This is likely to be a property of UCL rather than UCB simply being more conservative in its approach. As N (total number of arms) increases, the confidence bounds will decrease, whereas the confidence intervals have no natural tendency to decrease. Finally, the data in our sample are not normal; they follow a distribution that may be negative binomial or beta. While the UCB1 algorithm does not rely on an underlying assumption of normality, both UCL algorithms do.

In support of the broader research community, we intend to make the data from these bandit experiments available online at pslcdatashop.org [20]. Given that seasonality affected the performance of both bandit algorithms, having real-world data may be useful for others who seek to contribute more reliable bandit algorithms for UI optimization.

Implications

Implications for Practical Implementation: For those considering the use of bandits in the real world, we highly recommend using approaches that involve some degree of randomization (such as epsilon-greedy or Thompson Sampling). Without any randomization for comparison, there is no "ground truth" that would allow one to discover issues with seasonality, etc. As of this writing, there are now a variety of online resources and companies that can guide the implementation of bandits [4,37].

Implications for Data-Driven Designers: This work is intended to help designers understand the dangers of automated experimentation, in particular, how easy it is to optimize for the wrong metric. Indeed, it is not that maximizing engagement (our metric) is wrong, per se; however, when maximized to the extreme, it produces designs that appear quite bad.

Bandits are very capable of optimizing for a metric – but if this is not the best measure of optimal design, the bandit can easily optimize for the wrong outcome. For example, it is much easier to measure student participation than it is to measure learning outcomes, but conditions that are most highly engaging are often not the best for learning (e.g., students perform better during massed practice, but learn more from distributed practice [19]). In our study, the extremely large ship was the most engaging, but was unlikely to create the best learning environment [29]. Further work will continue to be necessary to refine our outcome criterion.

With the general increase in data-driven design, we think it is important for designers develop a *critical and dialectical* relationship to optimization metrics. To support the role of human judgment in an AI-Human design process, we recommend making it as easy as possible for designers to personally experience design variations as they are optimized. Metrics alone should not be the sole source of judgment; human experience will remain critical. Human designers should be trained to raise alternative hypotheses to understand why designs might optimize metrics but be otherwise objectionable [22].

When design becomes driven by metrics, designers must be able to participate in value-based discussions about the relationships between quantitative metrics and ultimate organization value [9]. Designers must be prepared to engage in an ongoing dialogue about what "good design" truly means, within an organization's value system. Design education should support training to help students engage purposefully with the meaning behind quantitative metrics.

Implications for AI-Assisted Design: In general, we wonder: how might people and AI work together in a design process? According to Norman's "human-technology teamwork framework" [35], human and computer capabilities should be integrated on the basis of their unique capabilities. For instance, humans can excel at creating novel design alternatives and evaluating whether the optimization is aligned with human values. AI can excel at exploring very large design spaces and mining data for patterns. Integrated Human-AI "design optimization teams" are likely to be more effective at optimizing designs than human or AI systems alone.

Importantly, both human judgment and AI-driven optimization can be wrong—so, ideally systems should be designed to support effective human-technology teamwork [35]. For instance, the original design of *Battleship Numberline* had a target size of 95% accuracy; while the designer felt this was best, this level of difficulty turns out to be significantly less engaging than other levels. At the same time, when the automated system was permitted to test a very broad range of options, it ended up generating designs that deeply violated our notion of good.

Designers can support optimization systems by producing "fuzzy designs" instead of final designs, where a fuzzy design is the range of acceptable variations within each design factor. More than a range, however, we recommend that designers deliver an "assumed optimal" setting for each design parameter along with a range of acceptable alternatives. AI can learn which alternative produces the best outcomes, but at the same time, designers can learn by reflecting on discrepancies between assumed optimal designs and actually optimal designs. This reflection has the

potential to support designer learning and create new insights for design and theory.

Implications for AI-Assisted Science: We note that previous work [31] showed that the effect of "surface-level" design factors (e.g., time limits, tickmarks, etc) may be mediated by "theory-level" design factors (e.g., "difficulty" or "novelty"). Thus, generalizable theoretical models might be uncovered algorithmically, or, more likely, through AI-human teams. Nevertheless, we anticipate significant opportunities for AI to support the discovery of generalizable theoretical models that can support both product optimization and scientific inquiry.

To be clear, the explosion of experiments with consumer software is driven by optimization needs, not science. That is, the purpose is to improve outcomes, not to uncover generalizable scientific theory. Yet, large optimization experiments have the potential to lead to generalizable insight (as with [29]). If the number of software experiments continues to increase (particularly with bandit-like automation), it would be wise to understand opportunities for how these experiments can also inform basic science. In areas like education, where millions of students are now engaged in digital learning, there may be many mutual benefits from a deeper integration of basic science (i.e., improving theory) with applied research (i.e., improving outcomes).

Online studies can easily involve thousands of subjects in a single day [28,38]. This is like having thousands of subjects show up to a psychology lab every day. Clearly, scientists don't have enough graduate students to analyze the results of dozens of experiments run every day of the year. This suggests that, while there is significant "Big Science" potential in conducting thousands of online experiments, deeper AI-human collaboration may be required to realize this potential. As others have suggested, bandits may help support this scientific exploration [27].

Yet, AI-Assisted experimentation may present some degree of existential risk. We have already discussed how runaway AI might optimize the "wrong thing." Keeping a human in the loop can mitigate this risk. However, there is another long-term risk: if AI-human systems are able to conduct thousands of psychological experiments with the intent of optimizing human engagement, might this eventually lead to computer interactions that are so addictive that they consume most people's waking hours? (Oops, too late!) Still, if online interfaces are highly engaging now, we can only imagine what will come with AI-assisted design.

Ethical considerations of online research

There are significant ethical issues that accompany large-scale online experiments involving humans. For instance, the infamous "Facebook Mood Experiment" [23] prompted a global uproar due to a perceived conflict between the pursuit of basic scientific knowledge and the best interests of unknowing users. Many online commenters bristled at the idea that they were "lab rats" in a large experiment. Online scientific research in education, although offering enormous potential social value (e.g., advancing learning science), faces the potential risk of crippling public backlash. We suggest that multi-armed bandits in online research could actually help assuage public fears.

First, bandit algorithms might help address the issue of fairness in experimental assignment. A common concern around education experiments is that they are unfair to the half of students who receive the lower-performing educational resource. Bandits offer an alternative where each participant is most likely to be assigned to a resource that brings better or equal outcomes. Indeed, Bandit-based experiments like ours are designed to optimize the value delivered to users, unlike traditional experimentation. Thus, we suggest that bandits may have a moral advantage over random assignment if they can adequately support scientific inference while also maximizing user value. Future work, of course, should explore this further.

CONCLUSION

The purpose of this paper is to illustrate how user interface design can be framed as a multi-armed bandit problem. As such, we provide experimental evidence to illustrate the promise and perils of automated design optimization. Our two large-scale online meta-experiments showed how multi-armed bandits can automatically test variations of an educational game to maximize an outcome metric. In the process, we showed that bandits maximize user value by minimizing their exposure to low-value game design configurations. By automatically balancing exploration and exploitation, bandits can make design optimization easier for designers and ensure that experimental subjects get fewer low-performing experimental conditions. While the future is promising, we illustrate several major challenges. First, optimization methods lacking randomization have serious potential to produce invalid results. Second, automatic optimization is susceptible to optimizing for the "wrong" thing. Human participation remained critical for ensuring that bandits were optimizing the "right" metric. Overall, bandits appear well positioned to improve upon random assignment when the goal is to find "the best design", rather than measuring exactly how bad the alternatives are.

ACKNOWLEDGMENTS

Many thanks to Allisyn Levy & the Brainpop.com crew! For intellectual input, thank you to Burr Settles, Mike Mozer, John Stamper, Vincent Aleven, Erik Harpstead, Alex Zook and Don Norman. The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305C100024 to WestEd, and by Carnegie Mellon University's Program in Interdisciplinary Education Research (PIER) funded by grant number R305B090023 from the US Department of Education. Additional support came from DARPA contract ONR N00014-12-C-0284.

REFERENCES

1. Agarwal, D., Chen, B.C., and Elango, P. (2009) Explore/Exploit Schemes for Web Content Optimization. *Ninth IEEE International Conference on Data Mining*, 1–10.
2. Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002) Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 235–256.
3. Berry, D. (2011) Adaptive Clinical Trials: The Promise and the Caution. *Journal of clinical oncology : official journal of the American Society of Clinical Oncology 29*, 6. 603–6.
4. Birkett, A. (2015) When to Run Bandit Tests Instead of A/B/n Tests. http://conversionxl.com/bandit-tests/
5. Bostrom, N. (2003). Ethical issues in advanced artificial intelligence. *Science Fiction and Philosophy: From Time Travel to Superintelligence*, 277-284.
6. Brezzi, M. and Lai, T.L. (2002) Optimal learning and experimentation in bandit problems. *Journal of Economic Dynamics and Control 27*, 1. 87–108.
7. Card, S., Mackinlay, J., & Robertson, G. (1990). The design space of input devices. *ACM CHI*
8. Chapelle, O., & Li, L. (2011). An empirical evaluation of thompson sampling. In*Advances in neural information processing systems* (pp. 2249-2257).
9. Crook, T., Frasca, B., Kohavi, R., & Longbotham, R. (2009, June). Seven pitfalls to avoid when running controlled experiments on the web. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1105-1114). ACM.
10. Drachen, A. and Canossa, A. (2009) Towards Gameplay Analysis via Gameplay Metrics. *ACM MindTrek,* 202–209.
11. Fogarty, J., Forlizzi, J., and Hudson, S.E. (2001) Aesthetic Information Collages: Generating Decorative Displays that Contain Information. *ACM CHI*
12. Gajos, K., & Weld, D. S. (2005). Preference elicitation for interface optimization. *ACM UIST* (pp. 173-182).
13. Gajos, K., Weld, D., and Wobbrock, J. Decision-Theoretic User Interface Generation. *AAAI*, (2008), 1532–1536.
14. Gittins, J. (1979) Bandit Processes and Dynamic Allocation Indicies. *Journal of the Royal Statistical Society*. Series B., 148–177.
15. Glaser, R. (1976). Components of a psychology of instruction: Toward a science of design. *Review of Educational Research*, 46(1), 1–24.
16. Hacker, S. (2014) *Duolingo: Learning a Language While Translating the Web.* PhD Thesis, Carnegie Mellon University School of Computer Science. May 2014
17. Hauser, J.R., Urban, G.L., Liberali, G., and Braun, M. (2009) Website Morphing. *Marketing Science.* 28, 2, 202–223.
18. Khajah, M., Roads, B. D., Lindsey, R. V, Liu, Y., & Mozer, M. C. (2016). Designing Engaging Games Using Bayesian Optimization, ACM CHI.
19. Koedinger, K. R., Booth, J. L., Klahr, D. (2013) Instructional Complexity and the Science to Constrain It *Science*. 22 November 2013: Vol. 342 no. 6161 pp. 935-937
20. Koedinger, K. R., Baker, R. S., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A data repository for the EDM community: The PSLC DataShop. *Handbook of educational data mining*, *43*.
21. Kohavi, R., Longbotham, R., Sommerfield, D., and Henne, R.M. (2008) Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery 18*, 1 140–181.
22. Kohavi, R., Deng, A., Frasca, B., Longbotham, R., Walker, T., and Xu, Y. (2012) Trustworthy Online Controlled Experiments: Five Puzzling Outcomes Explained. *KDD*
23. Kramer, Adam DI, Jamie E. Guillory, and Jeffrey T. Hancock. (2014) Experimental evidence of massive-scale emotional contagion through social networks. *PNAS*
24. Lai, T. (1987) Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics*; 15(3):1091–1114.
25. Lai, T., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics, 6,* 4–22.
26. Li, L., Chu, W., Langford, J., & Schapire, R.E. (2010) A Contextual-Bandit Approach to Personalized News Article Recommendation. *WWW*
27. Liu, Y., Mandel, T., Brunskill, E., & Popovic, Z. (2014) Trading Off Scientific Knowledge and User Learning with Multi-Armed Bandits. *Educational Data Mining*
28. Liu, Y., Mandel, T., Brunskill, E., & Popovi, Z. (2014) Towards Automatic Experimentation of Educational Knowledge. *ACM CHI*
29. Lomas, D., Patel, K., Forlizzi, J. L., & Koedinger, K. R. (2013) Optimizing challenge in an educational game using large-scale design experiments. *ACM CHI*
30. Lomas, D. Harpstead, E., (2012) Design Space Sampling for the Optimization of Educational Games. *Game User Experience Workshop, ACM CHI*
31. Lomas, D. (2014). *Optimizing motivation and learning with large-scale game design experiments* (Unpublished

Doctoral Dissertation). HCI Institute, Carnegie Mellon University. DOI: 10.13140/RG.2.1.5090.8645

32. Lomas, D., (2013). *Digital Games for Improving Number Sense* Retrieved from https://pslcdatashop.web.cmu.edu/Files?datasetId=445
33. Maclean, A., Young, R. M., Victoria, M. E., & Moran, T. P. (1991). Questions, Options, and Criteria: Elements of Design Space Analysis. *Human Computer Interaction*, *6*, 201–250.
34. Manzi, J. (2012). *Uncontrolled: The surprising payoff of trial-and-error for business, politics, and society*. Basic Books.
35. Norman, D. (in preparation) Technology or People: Putting People Back in Charge. Jnd.org
36. Scott, S. (2010) A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 639–658.
37. Scott, S. (2014) Google Content Experiments https://support.google.com/analytics/answer/2844870?hl=en&ref_topic=2844866
38. Simon, H. (1969). *The Sciences of the Artificial*. *C*Scott, S. L. (2015). Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry*, *31*(1), 37-45.
39. Stamper, J., Lomas, D., Ching, D., Ritter, S., Koedinger, K., & Steinhart, J. (2012) The rise of the super experiment. *EDM* p. 196–200
40. Stampfer, E., Long, Y., Aleven, V., & Koedinger, K. R. (2011, January). Eliciting intelligent novice behaviors with grounded feedback in a fraction addition tutor. In *Artificial Intelligence in Education* (pp. 560-562). Springer Berlin Heidelberg.
41. Vermorel, J. & Mohri, M. (2005) Multi-armed bandit algorithms and empirical evaluation. *Machine Learning: ECML 2005*, 437–448.
42. Yannakakis, G. N., & Hallam, J. (2007). Towards optimizing entertainment in computer games. *Applied Artificial Intelligence*, *21*(10), 933-971.