Multi-armed Bandits

"Multi-armed bandit experiments in the online service economy." *Applied Stochastic Models in Business and Industry* 31.1 (2015): 37-45.

Steven L. Scott

Sr. Economic Analyst
Google

October 29, 2019

Overview

1. Introduction and Background
2. Overview of Multi-armed Bandits
3. Thompson Sampling
4. Practical Considerations
5. Two Examples
6. Conclusion

1 Introduction and Background

2 Overview of Multi-armed Bandits

3 Thompson Sampling

4 Practical Considerations

5 Two Examples

6 Conclusion

Introduction - why should we use sequential allocation

Sequential Allocation

Use data we have gathered thus far in an experiment to drive our allocation strategy

- Multi-armed bandits with sequential allocation are well-equipped to deal with the challenges and differences of the service economy
- Randomization schemes can be made flexible with statistical modeling
- A/B testing can be made more efficient (faster) and economical (lower opportunity cost), without sacrificing statistical power

What's Different for the Service Economy

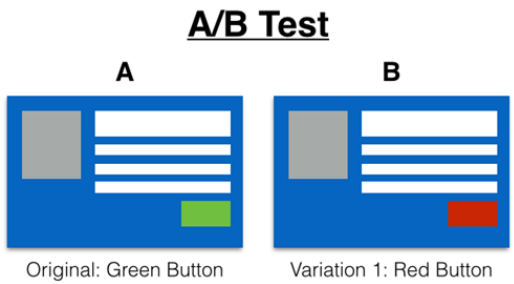
- A lot of experimental design discussion has been around manufacturing and agricultural experiments
  - What about companies like Google, Amazon, and Yahoo?
- Cost structures are different
- Data monitoring schemes are different

Why Change

- Type I (false positive - not that bad) versus Type II errors (false negative - more costly)
  - What type of error are we controlling for?
- Better utilize data streams
  - Update allocation strategy as experiment progresses
- Conversion rates can be low or similar (or both)
  - Difficult difference detection delays decisions
  - More on this later

A/B Testing

- Experimental technique to change content on websites, marketing pieces, and other forms
- Frequently used to compare a current product version with a new design



Why Should We A/B Test

- We need to experiment with our web design, but we need to make sure the viewers of the new design match our customer profile
  - We can use our customers or visitors as the experimental units
- We can collect evidence about what works and what does not
  - This can be extremely useful when we need to make other design decisions later on
- We can determine the best product to show our visitors, helping us to make the most out of our web traffic

Questions for A/B Testing

- We may have millions of visitors to a webpage during the day, which version should we show them?
  - How can we decide which version to show them?
    - ★ Should we consider the timing of the visit?
    - ★ Should we consider the country the visitor is from?
- When can we stop the testing and declare a win / lose / draw?

Issues with Modern A/B Testing

- Modern A/B testing 'best practice' utilizes the one factor-at-a-time framework (p. 43)
- This framework severely limits our ability to perform efficient and economical experiments
  - Need to run many similarly sized experiments to obtain the same precision as a well-designed multivariate experiment
- If there are significant interactions between experimental factors, this framework becomes even worse
  - How are we to measure the interaction effects?

1 Introduction and Background

2 Overview of Multi-armed Bandits

3 Thompson Sampling

4 Practical Considerations

5 Two Examples

6 Conclusion

Idea of Multi-armed Bandits

- Extend the A/B testing paradigm to AB...K testing in a sequential framework
  - Which arm to choose?
  - When to choose?
- Try to choose the arm that will produce the largest reward for each run
  - Conversions, clicks, subscriptions, etc.

Formal Description of Multi-armed Bandits

- Suppose we have K arms, with arm a associated with some quantity v_a giving the value of playing that arm
- We can assume the rewards come from some probability distribution $f_a(y|\theta)$
  - a - action taken
  - y - observed reward
  - θ - set of unknown parameters to be learned through experimentation
- Value $v_a(\theta)$ is a known function of the unknown θ
  - If θ were observed, the optimal arm would be known

Some Bandit Examples

Example 1: The binomial bandit
We have $\theta = (\theta_1, \ldots, \theta_k)$, a vector of success probabilities for K **independent** binomial models, with $v_a(\theta) = \theta_a$.

Example 2: Two factor experiment
We wish to investigate two factors (e.g. button color and button position), using dummy variables X_c for color and X_p for position. θ may be determined using a logistic regression:

$$logit\ P(conversion) = \theta_0 + \theta_1 X_c + \theta_2 X_p + \theta_3 X_c X_p$$

Then, $v_a(\theta) = logit^{-1}(\theta^T \mathbf{x_a})$

Example 3: Restless bandits
We could also model restless bandits, by assuming that the some of the coefficients from **Example 2** were indexed by time in a Gaussian process:

$$\theta_{t+1} = \mathcal{N}(\theta_t, \Sigma_t)$$

Why not be Greedy

- After n runs, we could find some estimate $\hat{\theta}$, and plug it into $v_a(\theta)$
  - Why not choose $\hat{a} = argmax_a v_a\left(\hat{\theta}\right)$?
- Estimation error - we could be wrong
- Limits opportunities to explore other arms
  - Explore / exploit trade-off
    - ★ More on this later

1 Introduction and Background

2 Overview of Multi-armed Bandits

3 Thompson Sampling

4 Practical Considerations

5 Two Examples

6 Conclusion

Thompson Sampling - Introduction

Thompson sampling is a heuristic we can use to allocate experimental units across possible arms.

Let $\mathbf{y_t}$ be the set of data observed up to time t, and let

$$\begin{aligned} w_{at} &= P(\textit{arm } a \textit{ is optimal} | \mathbf{y_t}) \\ &= \int I(a = \textit{argmax } v_a(\theta)) p(\theta | \mathbf{y_t}) d\theta \end{aligned}$$

where $p(\theta|\mathbf{y_t})$ is the Bayesian posterior distribution of θ given the data observed up to time t.

We assign the observation at time $t+1$ to arm a with probability w_{at}.

Thompson Sampling - Procedure

Suppose we are performing an experiment investigating K arms. How can we use Thompson Sampling to allocate arms at time $t+1$?

1. At time t during an experiment, update experimental data to obtain $\mathbf{y_t}$
   - e.g. Conversions, user characteristics, date / time information
2. Calculate $w_{at} = P(\textit{arm } a \textit{ is optimal} | \mathbf{y_t})$
3. At $t+1$, assign observations to arm a with probability w_{at}

Exploration

Let $a_t^* = argmax_a w_{at}$ be the arm which is most likely to be optimal at time t. Notice that the fraction of data committed to exploring at $t+1$ is $1 - w_{a_t^* t}^*$.

Thompson Sampling - Implementation and Benefits

- We can compute w_{at} using a Monte Carlo sample $(\theta^{(1)}, \ldots, \theta^{(G)}$ simulated from $p(\theta|\mathbf{y_t}))$

$$w_{at} \approx \frac{1}{G} \sum_{g=1}^{G} I\left(a = argmax\ v_a(\theta^{(g)})\right)$$

- Intuitive, simple, and generalizable
  - Suppose arm a has a 32% chance of being the best arm, it will then have a 32% chance of attracting the next observation
- Spreads observations across arms, allowing for probabilistically guided exploration
- Other strategies may include tuning parameters, making it somewhat more difficult to run an experiment

Thompson Sampling - Explore / Exploit

- A downside to the greedy approach was that it limited our ability to explore other arms
  - TS uses randomization, rather than a deterministic function
- With Thompson sampling, we can use $\gamma > 0$ to control the amount of exploration we do
  - Assign observations with probability proportional to w_{at}^{γ}
  - *Setting* $\gamma < 1$ *makes the bandit less aggressive, increasing exploration. Setting* $\gamma > 1$ *makes the bandit more aggressive.* (p. 39)

Thompson Sampling - Economic and Statistical Considerations

- Major concern in web experiments is losing conversions by showing users an inferior product
  - TS offers a better experience, by lowering the chance of a user seeing an inferior design
- Produces greater sample sizes for those arms near the top of the value scale
  - Extra data helps separate *good* arms from the *best* arms more quickly
- *Thompson sampling tends to shorten experiments while simultaneously making them less expensive to run for longer durations.* (p. 39)

Thompson Sampling - Randomization

- Thousands of visits to a website can occur before we can update w_{at}
  - Would prefer to update automatically, but technical delays may prohibit us
- TS randomly spreads observations across arms according to w_{at} while waiting for updates
- Nonrandomized algorithms do not, picking a single arm and making the same 'bet' for each experimental unit

1 Introduction and Background

2 Overview of Multi-armed Bandits

3 Thompson Sampling

4 Practical Considerations

5 Two Examples

6 Conclusion

When Should We Stop the Experiment

- The method we used to compute w_{at} for Thomson Sampling allows us to calculate a metric for deciding when experiments should end.
- Let $v_*\left(\theta^{(g)}\right) = max_a v_a\left(\theta^{(g)}\right)$ be the maximum value available **within Monte Carlo draw** g, and let $v_{a_t^*}\left(\theta^{(g)}\right)$ be the value for the arm (at draw g) deemed best **across all Monte Carlo draws**.

Using $v_*\left(\theta^{(g)}\right)$ and $v_{a_t^*}\left(\theta^{(g)}\right)$, we can simulate from the posterior distribution of *regret* from ending the experiment at time t:

$$r^{(g)} = v_*\left(\theta^{(g)}\right) - v_{a_t^*}\left(\theta^{(g)}\right)$$

$r^{(g)}$ is usually 0, but is sometimes positive.

How can we use Regret

- A nice property of regret is that its units match the units of value (e.g. dollars, clicks, or conversions)
- We can use an upper quantile of the distribution (such as the 95^{th} percentile) to calculate the 'potential value remaining' (PVR)
  - PVR can be thought of as the value per play that might be lost if we end the experiment at time t
- If PVR falls below a certain business-appropriate threshold we can end the experiment

Unit Free Regret

We can also calculate a measure of regret which is unit free:

$$\rho^{(g)} = \frac{v_* \left(\theta^{(g)}\right) - v_{a_t^*} \left(\theta^{(g)}\right)}{v_{a_t^*} \left(\theta^{(g)}\right)}$$

which measures the percentage change from the current apparently optimal arm.

If we are unable to determine a business-appropriate threshold for PVR, we can instead stop when ρ falls below some level, say $\rho < 0.01$.

Flexibility Using Contextual Modeling

The binomial bandit model is simplistic, and assumes the observations are independent. This assumption may not hold if our website is visited by individuals from different countries, as activity coming from these countries will be at different times during the day.

Similarly, website visitors may exhibit different behavior depending on the day of week (e.g. research a pair of shoes you want during your lunch break, but buy them at home on the weekend).

Clearly, the assumption of independence fails under these (and many other) circumstances.

A Contextual Model Using Logistic Regression

We can modify the binomial bandit, and instead fit a logistic regression:

$$logit(p_{a|\mathbf{x}}) = \beta_{0a} + \beta^T \mathbf{x}$$

where $\mathbf{x}$ is a set of variables describing the context of the observation, $p_{a|x}$ is the success probability if arm a is played during context $\mathbf{x}$, β_{0a} is an arm-specific coefficient, and β is a set of coefficients for the contextual information to be learned during the experiment.

Utilizing this method gives us adjusted optimality probabilities, but it assumes that we **know the important contexts**.

Contextual Hierarchical Modeling

If we are unsure of the important contexts for our experiment, we can assume that contexts occur as random draws from a distribution of contexts, using a beta-binomial hierarchical model for example:

$$\theta_{at} \sim Beta(\alpha_a, \beta_a)$$

$$y_t | a \sim Bin(\theta_{at})$$

The parameters we learn are $\theta = \{\alpha_a, \beta_a : a = 1, 2, \ldots, K\}$, and the value function is $v_a(\theta) = \alpha_a/(\alpha_a + \beta_a)$.
This parameterization helps us guard against an early lucky streak, which is random variation at the θ_{at} level.

Benefits of the Contextual Model

If we have two arms (A and B), and A performs slightly better during the week, but B is much better during the weekend, it is quite possible for the binomial bandit to determine A to be the optimal arm prior to seeing any weekend traffic.

Using the contextual model gives less credit to arms that perform well when conversions are high and gives more credit to arms that perform well during times when conversions are low.

Essentially, the contextual model allows us to slow down the bandit, avoiding spurious convergence to an arm with a lucky streak during a high-conversion period.

A Personalized Extension to the Contextual Model

The two previous examples of using contextual models omitted interactions between the contextual variables and experimental factors.

We could consider situations where we might want to fit interaction effects between experimental factors and contextual variables. Perhaps one version of a site performs better in Canada, and another version performs better in Mexico.

Provided we have a fairly robust data set, we may even be able to personalize results down to the individual level.

Multivariate Testing

- We may have more than one experimental factor we want to test
  - Statisticians: ‘multi-way layout‘
  - Internet companies: ‘multivariate testing‘
- Typical experimental designs choose a setup to find the minimal number of rows in a design matrix
  - Done to reduce costs, but web cost structures are different!
- We can use a probit model to allow the bandit to learn parameters associated with the experimental factors

Practical Considerations for Multivariate Testing

- In an online experiment, it is possible to randomize over all possible combinations of experimental factors
- Using all possible combinations makes the bandit's life more difficult
  - More parameters to learn, it may take longer to find an optimal arm
- By combining classical experimental design theory, we can find a reduced set of main effects and interactions we wish to estimate

1 Introduction and Background

2 Overview of Multi-armed Bandits

3 Thompson Sampling

4 Practical Considerations

5 Two Examples

6 Conclusion

Example - Low Conversion Rate Problem

Going back to the A/B testing framework, suppose we have a current version which produces conversion rates of 0.1%, and a new version which produces conversion rates of 0.11%.

To detect a difference in the conversion rates (using a one-sided test) with $\alpha = 0.05$ significance level and 0.95 power, we would need to accumulate 4,540,536 observations.

Suppose our website gets 100 visitors per day, under the classical framework, the experiment will require roughly 125 years to complete.

The Bandit is (Usually) Quicker

- Scott runs a simulation to investigate how long it would take the bandit to finish
  - Assume 100 visits per day (i.e. 100 observations per data update)
  - True success probabilities ($p_c = 0.001$ and $p_n = 0.0011$) held constant across all simulations
  - End experiment when ρ fell below 1%
- The simulation found the correct version 84 / 100 times
- 29 of 100 simulations ended within a year
- Fewer lost conversions than the classical experiment
  - For every 10,000 observations assigned to the current version, we lose one conversion

Histograms from Simulation

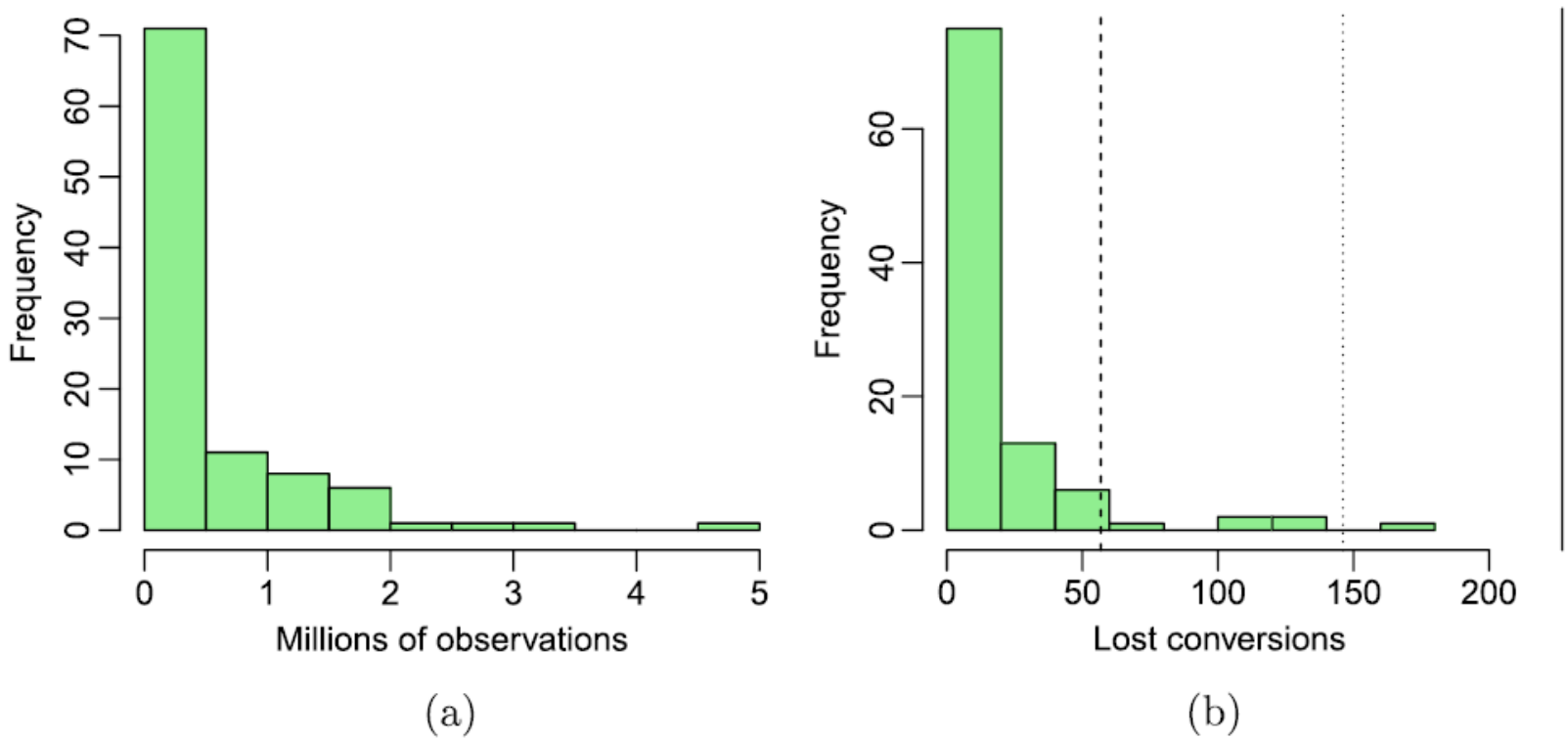


Figure: (a.) Histogram of the number of observations required to end the experiment in 100 runs of the binomial bandit. (b.) Histogram of the number of conversions lost during the experimental period. Vertical lines show the number of lost conversions under the classical experiment with 95% (solid), 50% (dashed), and 84% (dotted) power. (p. 41)

A More Complex Example from Google Analytics

- Scott also demonstrates the benefits of the bandit using a more complex example
- Suppose we have 6 arms, with 1 arm being the original version, and the other 5 being new variations
- If we use a significance level of .05 / (6-1), we find we need 15,307 observations in each arm
  - Uses the Bonferroni correction to control for multiple comparisons
  - Assuming 100 visitors per day, the experiment would require 919 days to complete

support.google.com/analytics/answer/2844870

A More Complex Example (cont'd)

Suppose in our experiment, the conversion rates are as follows:

Arm	Conversion Rate
1 (original)	4%
2	5%
3	4.5%
4	3%
5	2%
6	3.5%

Our original arm is beaten by two new designs: 2 (optimal) and 3 (suboptimal)

support.google.com/analytics/answer/2844870

A More Complex Example (cont'd)

Scott runs 500 simulations, assuming there are 100 visitors to a webpage each day (i.e. we observe 100 observations before each update). The simulation ends when an arm's optimality probability crosses 0.95 - this arm is declared the winner.

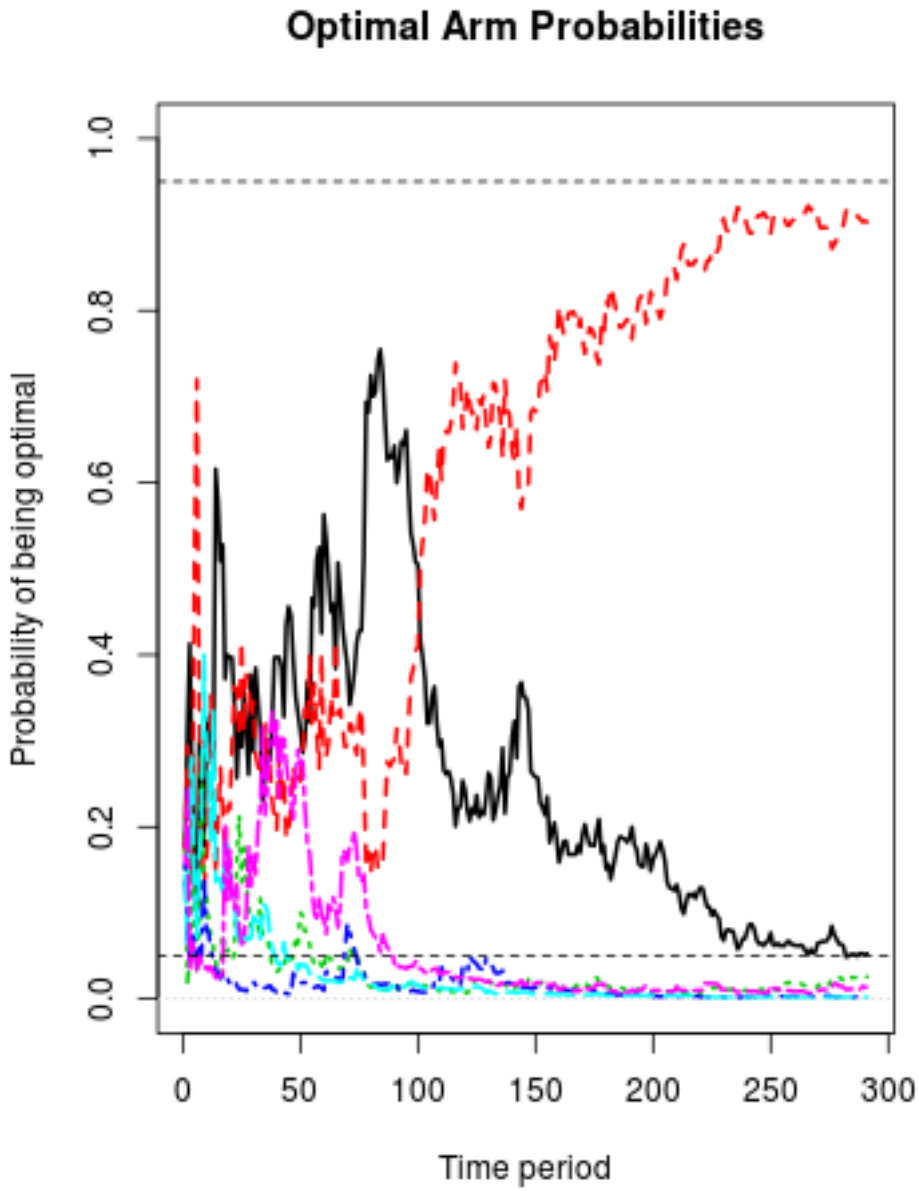
- Average experiment duration was 88 days, and the average number of saved conversions is 1,173
- *Even in the worst cases, running the experiment as a bandit saved over 800 conversions relative to the classical experiment*

support.google.com/analytics/answer/2844870

A More Complex Example (cont'd)

As part of his analysis, Scott includes a figure chronicling the history of one of the 500 simulations.

At first, the bandit seems confused, and we see a lot of variation in the "optimal" arm. We also observe a lucky run from the original version (black line). Finally, we see the optimal version (red line) pulling ahead and beating the original arm.



1 Introduction and Background

2 Overview of Multi-armed Bandits

3 Thompson Sampling

4 Practical Considerations

5 Two Examples

6 Conclusion

Multi-armed Bandits Control the Right Error

Type I Error: A/B Testing

Choosing a new version (e.g. button color, typeface, language) that is not significantly different than the original

Type II Error: A/B Testing

Failing to switch to a new version which is significantly better than the original

The multi-armed bandit allows us to focus on lowering economic opportunity cost, which will reduce our Type II Error rate.

Economical and Efficient Allocation

The bandit allocates in an economically and statistically efficient way:

- Economic: allocate to arms most likely to give a good return
  - Make sure we do not diminish user experience by showing bad experimental arms
- Statistical: allocate to arms we most want to learn about
  - Helps us separate the *good* from the *best*

Allocation can be made flexible as well, utilizing contextual modeling in cases where the binomial bandit's assumption of independence is violated.

The Bandit Helps Us Stop Experiments

- Along with the calculation of w_{at}, Thompson Sampling allows us to calculate regret
- Using $r^{(g)}$ or $\rho^{(g)}$ to determine when an experiment should stop gives us a flexible and intuitive stopping criterion
- Instead of waiting for a certain number of observations to be collected, we can use a more data-driven metric

Final Conclusions

- The bandit allows us to run full experiments faster
- Thompson sampling gives us an heuristic strategy we can use to allocate arms in an AB...K experiment
- By moving away from the considerations of typical experimental design methodology, we can use the bandit to focus on minimizing the costs associated with situations in the service economy