symmetry

MDPI

Article

Improving A/B Testing on the Basis of Possibilistic Reward Methods: A Numerical Analysis

Miguel Martín , Antonio Jiménez-Martín *, Alfonso Mateos and Josefa Z. Hernández

Decision Analysis and Statistics Group, E.T.S.I. Informáticos, Universidad Politécnica de Madrid, Campus de Montegancedo S/N, 28660 Boadilla del Monte, Spain; miguel.martin@alumnos.upm.es (M.M.); alfonso.mateos@upm.es (A.M.); josefaz.hernandez@upm.es (J.Z.H.)

* Correspondence: antonio.jimenez@upm.es

Abstract: A/B testing is used in digital contexts both to offer a more personalized service and to optimize the e-commerce purchasing process. A personalized service provides customers with the fastest possible access to the contents that they are most likely to use. An optimized e-commerce purchasing process reduces customer effort during online purchasing and assures that the largest possible number of customers place their order. The most widespread A/B testing method is to implement the equivalent of RCT (randomized controlled trials). Recently, however, some companies and solutions have addressed this experimentation process as a multi-armed bandit (MAB). This is known in the A/B testing market as dynamic traffic distribution. A complementary technique used to optimize the performance of A/B testing is to improve the experiment stopping criterion. In this paper, we propose an adaptation of A/B testing to account for possibilistic reward (PR) methods, together with the definition of a new stopping criterion also based on PR methods to be used for both classical A/B testing and A/B testing based on MAB algorithms. A comparative numerical analysis based on the simulation of real scenarios is used to analyze the performance of the proposed adaptations in both Bernoulli and non-Bernoulli environments. In this analysis, we show that the possibilistic reward method PR3 produced the lowest mean cumulative regret in non-Bernoulli environments, which proved to have a high confidence level and be highly stable as demonstrated by low standard deviation measures. PR3 behaves exactly the same as Thompson sampling in Bernoulli environments. The conclusion is that PR3 can be used efficiently in both environments in combination with the value remaining stopping criterion in Bernoulli environments and the PR3 bounds stopping criterion for non-Bernoulli environments.

Keywords: A/B testing; multi-armed bandit; stopping criterion; numerical analyses

Citation: Martín, M.; Jiménez-Martín, A.; Mateos, A.; Hernández, J.Z. Improving A/B Testing on the Basis of Possibilistic Reward Methods: A Numerical Analysis. *Symmetry* **2021**, *13*, 2175. https://doi.org/10.3390/sym13112175

Academic Editor: Bruno Apolloni

Received: 2 September 2021
Accepted: 3 November 2021
Published: 12 November 2021

1. Introduction

Currently, there is a major trend across different (retail, media, news, online advertising) companies to provide customers with the fastest possible access to the contents that they are most likely to use.

At the same time, companies that offer e-commerce services through their website or app are continuously optimizing the purchasing process so as to reduce customer effort to complete the online purchasing process and assure that the largest possible number of customers place their order.

The preferred approach for this and other types of service or user interface optimization solutions is to continuously make changes to the offered services or interfaces applying a specific indicator to measure which change produces the best expected indicator value.

This type of experimentation is commonly known as A/B testing. It is usually implemented by means of ad hoc in-house solutions or services or specialized software such as Google Analytics [1] or Optimizely [2].

The most widespread classical method for A/B testing is to implement the equivalent of RCT (randomized clinical trials): classical hypothesis testing techniques (usually *t*-Student with the Wald test) are applied to two randomized sub-samples (or as many samples as there are test variations) of customers to compute the number of experiments required for each sub-sample. The result is used to identify the best variation or define the maximum probabilities of type I and II error and the minimum expected effect as input values, if results are not statistically significant. Then either the variation with the highest expected value or the original solution will automatically be used depending on whether or not the test results are statistically significant. Further tests with other variations can be run continuously and iteratively.

Lately, some companies (like Google Analytics) have adopted a multi-armed bandit (MAB) problem-solving approach for their solutions [3–8]. They reused existing algorithms to deal with the exploration–exploitation trade-off. Known as *dynamic traffic distribution* in the A/B testing field [9], this technique calls for much fewer experiments. User guidelines and common errors arising when using these techniques in A/B testing are reported in [10].

Thanks to its sound performance processing delayed rewards, Thompson sampling [5] is the most popular MAB algorithm. However, it has the drawback that a priori knowledge of the type of distribution associated with the rewards or indicator is required. Purchase price, navigation time, number of pages visited before purchase or other factors may determine the distribution type. Unless action (purchase or content viewing) success or failure are measured by Bernoulli distributions, the distribution type may be unknown.

Another complementary option for optimizing A/B testing performance is the use of a better stopping criterion [11,12]. Bayesian techniques can be used for hypothesis testing to determine statistical significance. Again, however, Bayesian analysis depends on a priori knowledge of the rewards distribution type.

It has been found [13] based on recent numerical analyses that possibilistic reward (PR) methods [8] fare better than other MAB algorithms in specific delayed rewards scenarios, e.g., digital marketing content recommender systems. PR methods have the advantage that the reward distribution does not have to be known beforehand, although they have need of a stopping condition based on Bayesian or non-frequentist hypothesis testing. A/B testing can be carried out using the approximate rewards distribution function output by PR methods, governed by the stopping criterion.

The contributions of this paper are as follows:

- A performance comparison of the methods reported in the literature with a numerical analysis based on the precise simulation of real scenarios, as it is extremely difficult to evaluate all the methods in the real A/B testing configuration with statistical power. As far as we know, this is the first comparison for this type of problem.
- Apply PR methods, a recent family of MAB problem-solving methods, to A/B testing with dynamic traffic allocation technology. These methods have proved to outperform classical methods mainly in contexts where rewards are delayed and do not match a parametric distribution. These are common scenarios in many A/B testing configurations.
- Propose and develop a new stopping criterion for A/B testing configurations (independent of static or dynamic configurations) also based on PR methods and applicable in any usual A/B testing configuration.

The paper is structured as follows. Section 2 gives a brief description of A/B testing and improvements aimed at optimizing how tests are carried out (dynamic traffic distribution and stopping criterion). Section 3 briefly reviews possibilistic reward methods and their use in A/B testing. A new stopping criterion based on PR methods founded on both value remaining and confidence levels is introduced in Section 4. Section 5 describes the numerical analysis carried out and the results. Lastly, some conclusions and future research work are outlined in Section 6.

2. A/B Testing: Optimization of Digital Services and Content through Internet

It is common practice in companies that offer services and products through online channels (web and mobile apps) to continuously optimize their user interfaces with the aim of improving one or more of their key business indicators, such as customer satisfaction, online sales, content consumption times, or advertising conversion rates. Examples of sectors that perform this optimization are Internet travel agencies, the retail sector (clothing, food,...), content services (Netflix, HBO, Spotify...), news services or large multinational companies (Amazon, Google, Facebook...).

For optimization purposes, continuous experimentation is conducted on:

- The creation of different alternative products or services: changing the size, color and shape of user interfaces, altering the order of the purchase process steps, offering different contents, establishing different prices for a product, etc.
- The division of customers or users who access the service through web or digital channels into different groups, where each group is offered a version or variation of the experiment.
- The identification of the best variation according to a previously defined indicator, such as a conversion ratio in advertising, the number of times a button is clicked or a page is accessed, the browsing time, number of steps to make a purchase or the income earned.

These experiments are known in the industry as A/B testing, randomized controlled trials (RCT) where different variations are tested until there is statistical significance.

Therefore, according to the statistical techniques applied in RCT, A/B testing will randomly divide the set of customers/users in as many sub-samples (groups) as variations are to be tested. The minimum number of experiments to be executed to assure statistical significance must be defined a priori using classical hypothesis testing techniques. Then, each variation is shown to one group until the necessary number of experiments is reached, where each group should include the same number of experiments.

The most widespread techniques for identifying the minimum number of experiments to be executed to assure statistical significance (p-value $\leq \alpha$) are the Wald or t-Student tests, where the following input parameters must be provided:

- α, false positive rate, or type I error.
- β, true positive ratio, or (1—type II error).
- Minimum desired effect of the variation.

Note that the winning variation will be automatically configured for all customers/users until a new experiment is performed on the same service or content.

Two options are currently used to implement A/B testing:

- Ad hoc developments, mainly using proprietary software (primarily large content managers, such as Google, Netflix, Facebook, Amazon...), or libraries, such as Facebook Planout, and plug-ins by e-commerce platforms, such as Magento or Pentashop.
- Specialized experimentation software, where there is a wide variety of vendors, notably Google Optimizer, Optimizely, AB Tasty and VWO.

These products usually have a similar standard operating mode composed of the following steps:

1. A small javascript code snippet is inserted in the home page and in each page involved in the experimentation. This javascript library is responsible for implementing the different variations by modifying the html of each page.
2. One or more variations are made to the format of the web page to be changed using a tool that shows the current page and elements, such as menu options, button sizes, typeface, colors, or element arrangements on the web page to be changed.
3. The objective to be optimized is defined. Most products offer different types of objectives. The javascript code snippet inserted in the customer web is also responsible for reporting whether or not an objective and its value are achieved. Common objectives are number of clicks, conversions, sales or navigation time.
4. The statistical significance requirements are defined: α, effect size and statistical power.
5. The test is executed:
    - Traffic is distributed uniformly or using weights.

- If the results are statistically significant, the winning variation will become the default configuration until another experimentation or test is carried out.

Optimized testing depends on two optimization processes, namely dynamic traffic distribution and the stopping criterion. The latest A/B testing solutions include these processes as a means of achieving statistical significance by avoiding tests of the worst options to achieve the lowest opportunity cost.

2.1. Dynamic Traffic Distribution

In A/B testing, traffic is originally distributed equally for each of the variations to provide the same number of experiments. However, the percentage of traffic that each experiment is to receive can also be set manually. Nevertheless, it is more efficient to redistribute traffic dynamically, sending more or fewer experiments to variations that perform better or worse, respectively, provided that statistical significance is achieved.

MAB methods are a good alternative for dynamic traffic distribution, choosing the option (arm) that is displayed when a user/customer views a website. This will send a scenario-dependent delayed stochastic reward (Bernoulli or other distributions). The implemented option selection strategy should minimize the opportunity cost, thereby optimizing the expected reward value.

A review of the most important strategies for the MAB problem can be found in [14], and a numerical study on the basis of five complex and representative scenarios was conducted and reported in [8] to compare their performances.

MAB-based dynamic traffic distribution solutions for measuring action success or failure (Bernoulli distributions) are:

1. Thompson sampling adapted for Bernoulli rewards based on probability matching. This method weights the probability of the expected value of an option being greater than the others. This weight (or selection probability) is used to choose which option from a random sample is most likely to be the best for execution. This weight is usually calculated applying simulation techniques, which have high computational costs. Therefore, weights are only updated every N hours or N options. Thompson sampling is the most popular MAB-based dynamic traffic distribution technique on the market, as it has very good performance, even within commercial A/B testing systems where there is a time delay between an option and its feedback [13]. Google, with Google analytics (although the current version for experimentation, Google Optimize, does not yet support MBA algorithms), VWO [15], and ABTasty [16] are some of the companies adopting this approach.
2. Adapted e-greedy algorithms. This is a simpler algorithm, offering linear rather than logarithmic convergence, which calculates the best options at regular intervals or after N iterations. The algorithm then distributes 80% of the traffic to the best options (exploitation), whereas the other 20% is distributed to all options (exploration) for the duration of the iteration. Adobe Target [17] is an example of this solution. Adobe Target uses Bernstein's confidence bounds to run an hourly check of the best two options. These two options then receive 80% of the traffic.
3. Although not reported, some large companies implementing their own ad hoc experimental software may very well be using a more efficient form of Thompson sampling, with dynamic algorithm updating at each decision. Otherwise, there is little information in the literature on how dynamic traffic distributions within non-Bernoulli distribution scenarios are performed. We think that this is the option adopted by important content managers since the algorithm is well known, and it works optimally in scenarios under Bernoulli delayed rewards.

Thompson sampling is not an option for experiments within a non-Bernoulli scenario, that is, measuring browsing time, number of visited pages, total revenue, etc., because the reward distribution cannot be parameterized.

Even in some cases where the reward distribution can be parametrized on one family of distributions, it will be necessary, if there is no simple way to compute the a posteriori distributions (*conjugate priors*, for example), to apply simulation techniques. This can make the computation intractable or computationally very expensive. Therefore, the major companies, such as Adobe and ABTasty, use other alternatives, mainly a variation on e-greedy algorithms.

As pointed out in [18], "Other vendors such as Google, with Google Analytics, and ABTasty, do not provide information on whether or not and how they perform dynamic traffic distribution with objectives not following a Bernoulli distribution".

However, PR methods are an alternative in this context because no knowledge of the distribution type associated with the rewards or indicator for optimization is required beforehand. On this ground, we put forward an alternative PR method-based dynamic traffic distribution for A/B testing within non-Bernoulli reward scenarios.

2.2. Stopping Criterion

The stopping criterion plays a key role in the execution of A/B testing experiments. It is used to decide when a variation is considered to be the best. It stops the traffic to the other variations and automatically sets up this variation as the default configuration until the next experiment or test is carried out.

The de facto method used to define the stopping criterion in most approaches is based on a classical hypothesis test, where:

- The variation existing before the test is performed is considered as the default variation, and the null hypothesis establishes that the new variations are not better than the default configuration.
- The number of samples required for each variation to assure statistical significance is computed before performing the test. The false-positive, false-negative rates and minimum detectable effect also have to be established. The most used hypothesis tests are the Ward, *t*-student or χ^2 tests.
- Then, the test is run with the samples for each variation.
- If the test does not reject the null hypothesis, then the default configuration continues to be used. Otherwise, we will use one of the alternative variations as the best.
- All the traffic will be diverted from the default variation to the best variation.

Note that if there are more than two variations, adjustments, such as the Bonferroni correction [19], are usually made to counteract the increase in the probability of type I errors.

However, these classic stopping criteria are not very efficient, since they are unable to dynamically stop the test when there is enough evidence to suggest that one variation is better than the others [20].

Recently, the most innovative companies are introducing other dynamic stopping criteria to reduce testing costs, whereby they arrive at the same statistical significance in a similar way. These new methods, although perfectly applicable to classical A/B testing, come hand in hand with the new methods for dynamic traffic distribution. The multi-armed bandit paradigm is the most popular, since the number of samples that have to be executed for each variation is determined dynamically rather than using classical hypothesis tests to identify the number of samples required to achieve statistical significance.

These new criteria are based on different approaches (Bayesian, inequalities bounds...). We will now review the most important approaches. However, we should first establish the configuration criteria for proper comparison in the numerical analysis section.

When defining a stopping condition, it is important to define the conditions for a test to be considered positive. To do this, a parameter nomenclature should be defined, and standard terminology should be used across the different methods. For clarity's sake, we will try to stick to the classical nomenclature:

- α, this value represents the maximum number of false positives or the type I error percentage threshold considered acceptable when identifying the winning variation. The lower this value is, the later the test stops. Accordingly, $(1-\alpha)$ will be the confidence threshold that we consider adequate to consider the default variation as the best.

- β, this value represents the maximum number of false negatives or the type II error percentage threshold that we consider acceptable when identifying the winning variation. The lower this value is, the later the test stops. Accordingly, $(1-\beta)$ will be the confidence threshold that we consider adequate to consider one of the alternative variations as the best.
- MED (minimum detectable effect), this value is closely related to parameter β. It indicates the percentage deviation from the target of the default variation required for an alternative variation to be considered better.

The stopping criteria most commonly used by leading manufacturers are currently based on one of the following two approaches:

- a first approach is based on a Bayesian approach [1,20], where N simulations are executed every x hours. In each one, the expected value of each variation is sampled according to its density function (this density function corresponds to a beta function in the case of Bernoulli rewards). Then, the so-called *value remaining* is computed for each simulation according to

  $$value\_remaining = (v_{max} - v^*)/v^*,$$

  where v_{max} is the best variation across the simulations and v^* is the value of the best variation in the current simulation.

  The next step is to compute the percentile p_{thres} (usually 95 %) for the value remaining across all simulations. The test ends when p_{thres} is lower than parameter ρ_{min}.

  ABTasty and VWO are based on this approach, and Google is one of the vendors that used this stopping criterion. Note that this stopping method is only practicable if the family of probability distributions to which the rewards belong is known a priori, so that the uncertainty of their parameters and, consequently, the probability function of the expected value of each reward can be modeled using a Bayesian approach. This is immediate in Bernoulli distributions, but we do not know how vendors like Google Analytics implement this stopping condition for the other objective. It is probable, therefore, that Google Analytics only implements MAB with this dynamic stopping condition in situations with Bernoulli distributed rewards.

  For this test to be comparable with classical hypothesis tests, we have to account for the fact that this approach does not consider null or alternative hypotheses; instead one hypothesis is considered to be better than another if there is a confidence interval. In order to comply with the above terminology, we would have to make the following distinction:

  - α corresponds to the value (1-p_{thres}) as long as the default variation is the winning variation.
  - β corresponds to the value (1-p_{thres}) as long as the winning variation is any of the new variations.

  The most popular configuration for this approach is to use the same value for parameters α and β, usually 0.05. Besides, MED matches parameter ρ_{min}.

  Algorithm 1 describes the method for Bernoulli distributed rewards. Briefly, this algorithm works as follows. First, a vector is initialized containing the number of wins, i.e., the number of times each variation comes out on top across all the simulations (lines 1–3). Second, a sample is iteratively generated with the number of successes and failures of each variation of a Beta distribution for each simulation, and the counter of the number of wins of the variation with the largest sample is incremented (lines 4–10). Third, for each of the N simulations, the value of its value remaining and the percentile related to the required confidence level for all the values remaining are computed (lines 11–16). Finally, we check whether or not this value is lower than the MED parameter to decide if the stopping criterion condition is met (lines 17–22).

Algorithm 1 Value remaining stopping criteria

```
Require: ($N$: number of simulations, $success$, $fail$: arrays including the number of success
    and fails rewards of each variation, respectively;)
Ensure: (True/False)
    for $i \in \{1,...,num\_variations\}$ do
        $best\_arm[i] \leftarrow 0$
    end for
    for $i \in \{1,...,N\}$ do
        for $j \in \{1,...,num\_variations\}$ do
            $sims[i][j] \sim Beta(1 + success[j], 1 + fail[j])$
        end for
        $max\_var \leftarrow argmax(sims[i])$
        $best\_arm[max\_var] \leftarrow best\_arm[max\_var] + 1$
    end for
    $best\_arm\_of\_sims \leftarrow argmax(best\_arm)$
    for $i \in \{1,...,N\}$ do
        $max\_value \leftarrow max(sims[i])$
        $value\_max\_sims \leftarrow sims[i][best\_arm\_of\_sims]$
        $rvalue\_remain[i] \leftarrow (max\_value - value\_max\_sims)/value\_max\_sims$
    end for
    $\rho \leftarrow percentil(confidence\_level, value\_remain)$
    if ($\rho \leq MED$) then
        return True
    else
        return False
    end if
```

An alternative approach is where a stopping method based on confidence intervals computed by the Bernstein inequality [21] is used. A variation will be the winner when its 95% confidence interval does not overlap with the 95% confidence interval of any other variation [17]. Unlike other solutions, once the winning variation is known, Adobe will automatically redistribute 80% of the traffic to the winning option and the remaining 20% uniformly across all the variations, including the winner. If there was no winning option in the future, the traffic would be redistributed according to the e-greedy strategy explained above.

To compute the confidence intervals using the Bernstein inequality, the variances of the distributions have to be known. As these variances are unknown in most cases, we have to use a variation of this method that uses the sample variance.

Although the exact method used by any vendor is not known, the *empirical Bernstein* method is the most used in the literature [22]. This method computes the confidence intervals according to the following equation:

$$|\overline{\mu} - \mu| \leq \overline{\sigma}\sqrt{\frac{2\log(3/\delta)}{n}} + \frac{3R\log(3/\delta)}{t},$$

where $\overline{\mu}$ is the sample mean, μ the expected value, $\overline{\sigma}$ is the sampling standard deviation, δ is the significance level, usually 0.05, n is the number of samples and R is the range (maximum value - minimum value). Adobe Target is one of the vendors that uses this approach as a stopping criterion.

As above, if this test is to be comparable with classical hypothesis tests, we have again to take into account the following:

- Rather than testing a null or alternative hypothesis, this approach considers that one hypothesis is better than another if there is a confidence level. In order to comply with the terminology described above, we would have to make the following distinction:

  - α corresponds to the percentile value used to compute the confidence intervals, as long as the default variation is the winning configuration.
  - β corresponds to the percentile value used to compute the confidence intervals, as long as the winning variation is any of the new variations.

  Again the most used configuration in this approach is to use the same value for parameters α and β.

- MED appears to be set to 0 in this algorithm, detecting the smallest variation. However, a slight extension should be introduced for proper comparison with other criteria: the overlap with the worse variations should be considered as being less than or equal to MED rather than there being no overlap at all.

Algorithm 2 describes this stopping criterion. This basically computes the differences between the lower bound of the variation with the best empirical mean and the upper bound of each of the other variations. If any of these differences is less than the MED, the stopping criterion will have been reached.

Algorithm 2 Adobe stopping criterion

```
Require: ($num\_samples$ = number of samples of each variation, $means$, $vars$: arrays with the means and variances of each variation, respectively; $BB$ function that provides the $n$ percentile bound according to Bernstein's inequalities)
Ensure: (True/False)
$best \leftarrow$ argmax($means$)
$lcb\_best \leftarrow BB(1 - confidence\_level, num\_samples[i], means[best], vars[best])$
for $i \in \{1,...,num\_variations\}$ do
    if $i \neq best\_variation$) then
        $ucb \leftarrow BB(confidence\_level, num\_samples[i], means[i], vars[i])$
        if $lcb\_best$ - $ucb \leq$ MED) then
            return True
        end if
    end if
end for
return False
```

3. Possibilistic Reward Method in A/B Testing

In this section, we describe a new approach in which MAB PR methods [8,14] are used to perform dynamic traffic distribution in A/B testing. Table 1 shows a comparison of these methods with the techniques described in Section 2.

Table 1. Dynamic allocation methods comparative.

Dynamic Allocation Method	Type of Metrics	Performance
Thompson sampling and Probability matching	Just for metric with known parametric distribution. (i.e., Bernoulli, Normal)	Good. Minimize cumulative regret.
ϵ-greedy	For any type of metric distribution.	Medium. Good reduction of cumulative regret.
No dynamic allocation (classic A/B)	For any type of metric distribution	Bad. No reduction of cumulative regret.
PR2 and PR3	For any type of metric distribution.	Good. Minimize cumulative regret.

Possibilistic reward methods (PR1, PR2 and PR3) [8,14] have recently been proposed as alternatives to MAB algorithms in the literature. A review of the most important allocation strategies can be found in [14].

The basic idea of the PR1 method is as follows: the uncertainty about the arm expected rewards are first modeled by means of possibilistic reward distributions derived from a set of infinite nested confidence intervals around the expected value on the basis of the Chernoff–Hoeffding inequality [23].

Then, the method follows the *pignistic probability transformation* from decision theory and the transferable belief model [24]. The pignistic probability transformation establishes that when we have a plausibility function, such as a possibility function, and any further decision-making information, we can convert this function into an probability distribution following the *insufficient reason principle*.

Once we have a probability distribution for the reward of each arm, then a simulation experiment is carried out by sampling from each arm according to their probability distributions to find out which one has the highest expected reward. Finally, the selected arm is played and a real reward is output.

As mentioned above, the PR1 method starting point is the Chernoff–Hoeffding inequality [23]. This inequality establishes an upper bound on the probability that the sum of random variables deviates from its expected value for [0,1], which can be used to build an infinite set of nested confidence intervals.

The difference between PR1, PR2 and PR3 lies in the type of concentration applied and subsequent approximations. PR1 and PR2 are based on the Hoeffding concentration, whereas PR3 is based on a combination of the Chernoff and Bernstein concentrations.

A numerical study based on of five complex and representative scenarios was performed in [8] to compare the performance of PR methods against other MAB methods in the literature. PR2 and PR3 methods perform well in all representative scenarios under consideration, and are the best allocation strategies if truncated Poisson or exponential distributions in [0,10] are considered for the arms. Besides, Thompson sampling (TS), PR2 and PR3 perform equally with a Bernoulli distribution for the arm rewards. PR2 is exactly the same as the generalization of the TS method proposed in [25] (see Algorithm 2).

Moreover, the numerical analyses conducted recently in [13] show that *possibilistic reward* (PR) methods outperform other MAB algorithms in digital marketing content recommendation systems for campaign management, another scenario with delayed rewards.

Lastly, PR methods have one big advantage over other MAB algorithms, including TS: all they need to know is the interval to which the reward belongs rather than the total reward distribution. PR methods approximate a distribution function for rewards that can also be used to perform a classic A/B test, albeit with a stopping condition based on Bayesian and non-frequentist hypothesis tests. In this way, experimentation can be efficiently carried out with these methods in contexts where the objective is not confined merely to action success or failure (Bernoulli distribution), but also to the minimization of the total number of page views or the duration of a session, or the maximization of the total income from web e-commerce.

4. New Stopping Criteria Based on Possibilistic Rewards

In this section, we propose the adaptation of the stopping criterion used in both Google Analytics and Adobe Target to account for possibilistic reward methods.

4.1. PR2 and PR3 Value Remaining

The stopping method based on the *value remaining* used by Google Analytics [20] is very efficient in environments with rewards following a Bernoulli distribution, since it has to know the exact distribution of the expected rewards in order to carry out the simulations.

The distribution of the expected rewards is inferred with a Bayesian approach as follows: the reward distribution function is modeled by a parametric family. A prior distribution is applied to the parameters of this family: both the *a posteriori* distribution of the parameters and the expected rewards are inferred by Bayesian techniques.

This approach, however, has a drawback: the shape of the reward distribution has to be known or modeled by a family of parameterizable distributions on which priors can be applied. In addition, it should be tractable or at least computationally efficient to update the a posteriori distributions and the expected value. This is not very often the case in many real contexts, where the family to which the reward distribution belongs (normal, Poisson, Bernoulli) is unknown. Besides, if the distribution is known or can be modeled, it is very difficult to make an efficient inference using, at least it cannot be apply techniques as, for example, conjugate priors.

To overcome this problem, we propose the following approach. We can approximately model the probability distribution of the expected rewards efficiently by applying the possibilistic rewards methods (PR2 and PR3) for the reward in each variation. To do this, only the minimum and maximum reward bounds have to be known rather than the distribution of each reward. This information is commonly available in real contexts.

Once we have derived the density function of the expected reward (Step 3 in PR2 and PR3), we just apply the simulation and stopping condition techniques used in [20]. Algorithm 3 shows the process according to which these methods calculate the value remaining.

The steps executed by this algorithm and Algorithm 1 are identical, except for the manner in which the samples are generated across the simulations. Specifically, although it uses the Beta probability distribution, its parameters are computed according to the PR2 or PR3 specifications (lines 6–16).

In the following sections, reporting a numerical analysis of these methods, they will be denoted as *PR2 value remaining* and *PR3 value remaining*.

4.2. PR2 and PR3 Confidence Level

Emulating confidence level-based stopping criteria, such as empirical Bernstein in Adobe Target, we now propose a stopping criterion computed from approximations to the probability distributions of the expected reward derived from the PR2 and PR3 methods.

A comparison of the new stopping criteria with the conditions explained in Section 2 is shown in Table 2.

To do this, we need a function that outputs the percentile value, which will be used as a confidence level, from distributions PR2 or PR3. As PR2 and PR3 are Beta distributions, we can use the quantile function, also called *ppf* (percentile point function), to compute these dimensions. This function, denoted by *beta_ppf* from now on, can be analytically obtained and is available in any statistical software library.

Once these dimensions have been derived, we have practically the same stopping criterion as the one used by Adobe Target (see Algorithm 4). It operates like Algorithm 2, except that the bounds are now computed by means of the *beta_ppf* function of the Beta distribution (lines 3–5).

Table 2. Stopping criteria methods comparative.

Stopping Criteria Method	Type of Metrics	Performance
Thompson sampling value remaining	Just for metric with known parametric distribution. (i.e., Bernoulli, Normal)	Good. Minimize number of samples.
Bernstein bounds	For any type of metric distribution.	Medium. Good reduction of number of samples.
Classical hypothesis testing (*t*-Student)	For any type of metric distribution	Bad. No reduction of number of samples.
PR2 and PR3 value remaining	For any type of metric distribution.	Good. Minimize of number of samples.
PR2 and PR3 bounds	For any type of metric distribution.	Good. Minimize of number of samples.

Algorithm 3 PR value remaining stopping criteria

```
Require: ($N$: number of simulations, $sample\_means$, $sample\_vars$: sample means and variances of each variation reward, $num\_samples$: number of samples of each variation reward)
for $i \in \{1,...,num\_variations\}$ do
    $best\_arm[i] \leftarrow 0$
end for
for $i \in \{1,...,N\}$ do
    for $j \in \{1,...,num\_variations\}$ do
        if (method == PR2) then
            $\alpha \leftarrow sample\_means[j] \times num\_samples[i]$
            $\beta \leftarrow num\_samples[i] - \alpha$
        else ( # method==PR3)
            $max\_var \leftarrow sample\_means[j] \times (1 - sample\_means[j])$
            $confidence\_bound \leftarrow \sqrt{-\log 0.05/(2 \times num\_samples[j])}$
            $\sigma^2 \leftarrow min\{max\_var, sample\_vars[j] + confidence\_bound\}$
            $r \leftarrow num\_samples[j] \times \frac{max\_var}{\sigma^2}$
            $\alpha \leftarrow r \times sample\_means[j]$
            $\beta \leftarrow r - \alpha$
        end if
        $sims[i][j] \sim Beta(1 + \alpha, 1 + \beta)$
    end for
    $max\_var \leftarrow argmax(sims[i])$
    $best\_arm[max\_var] \leftarrow best\_arm[max\_var] + 1$
end for
$best\_arm\_of\_sims \leftarrow argmax(best\_arm)$
for $j \in \{1,...,N\}$ do
    $max\_value \leftarrow max(sims[i])$
    $value\_max\_sims \leftarrow sims[i][best\_arm\_of\_sims]$
    $remain\_value[i] \leftarrow (max\_value - value\_max\_sims)/value\_max\_sims$
end for
$\rho \leftarrow percentile(confidence\_level, remain\_value)$
if $(\rho \leq MED)$ then
    return True
else
    return False
end if
```

4.3. Testing the Null Hypothesis

Note that test A/B executions usually have two main purposes. The main objective is to test the statistical significance of two content delivery or user interface alternatives and have the best alternative, irrespective of whether it is the original or any variant, automatically deployed for further user interaction. The stopping criterion explained and proposed in this paper will be enough for this purpose.

Furthermore, the experimenter will want to know if the test arrives at reasonable scientific conclusions, that is, whether or not we can reject the null hypothesis. Note that any of the criteria mentioned in this paper (the value remaining, Adobe, PR value remaining, PR bound stopping criteria) will stop in both scenarios: null hypothesis rejection or non-rejection. Therefore, we will need to run an algorithm to check for null hypothesis rejection once the stopping criteria has been reached in order to arrive at the final conclusion for this approach. Algorithm 5 checks if the mean value of the best variation is greater than the mean value of the original variation plus the MED. If it is, we can reject the null hypothesis, otherwise it cannot be rejected.

Algorithm 4 PR bound stopping criteria

```
Require: ($num\_samples$: n. of samples of each variation, $means, vars$: means and variances
    of each variation, respectively)
$best \leftarrow argmax(means)$
$lcb\_best \leftarrow Beta\_ppf(1 - confidence\_level, num\_samples[i], means[best], vars[best])$
for $j \in \{1,...,num\_variations\}$ do
    if $(i \neq best\_variation)$ then
        ucb $\leftarrow Beta\_ppf(confidence\_level, num\_samples[i], means[i], vars[i])$
        if $(lcb\_best - ucb \leq MED)$ then
            return True
        end if
    end if
end for
return False
```

Algorithm 5 Check for null hypothesis rejection

```
$best\_mean \leftarrow max(means)$
if $(best\_mean - MED > default\_variation)$ then
    return "reject null-hypothesis"
else
    return "fail to reject null-hypothesis"
end if
```

5. Numerical Experiments

The numerical experiments conducted are divided into two phases. In a first phase, we tested the stopping criteria in order to discard any criteria whose performance was considerably below the level of the best criteria. Then, in the second phase, we apply simulation techniques in four scenarios accounting for Bernoulli and non-Bernoulli distributions to analyze the performance of the different methods (probability matching, Thompson sampling, e-greedy, classic A/B testing, PR2 and PR3) on the objectives to be optimized.

5.1. Testing Stopping Criteria

We first tested the stopping criteria existing in the literature and proposed in this paper with variations that had different density distributions in order to discard any whose performance was considerably below the level of the best criteria.

These stopping criteria depend exclusively on the density functions of the best and second-best variations and will return a worse result the closer their means and the more alike their density functions are. Thus, by directly recreating density functions of different variations (similar to the functions used in the Phase-2 simulation scenarios), we can quickly identify whether or not there is a stopping condition that behaves considerably worse than the others in the scenarios under consideration. It can then be discarded for the simulations, thus saving computational costs.

We considered three scenarios. First, we considered two Bernoulli variations with means 0.05 and 0.045, and equal traffic allocation. Next, we again considered the two Bernoulli variations with means 0.05 and 0.045, using in this case dynamic traffic allocation with a well-known standard method like Thompson sampling. Lastly, we considered ten non-Bernoulli variations with a mixture of a Bernoulli and exponential distribution in the last and most extreme scenario.

Table 3 shows the average number of resulting samples and the 95% confidence interval for ten replications in each scenario simulation. Note that the stopping condition in all these cases occurs when the MED value falls below the value 0.01.

Table 3. Number of samples for scenarios.

Stopping Criteria	Scenario 1	Scenario 2	Scenario 3
PR2 value remaining	**1045 [609, 2138]**	**7020 [3887, 13,642]**	84,303 [46,570, 163,181]
PR2 bound	9746 [6187, 15,958]	46960 [29,796, 76,815]	101350 [64,240, 165,538]
Empirical Bernstein	34,333 [28,294, 41,911]	133,265 [109,786, 162,613]	232,184 [191,208, 283,198]
PR3 value remaining	-	-	48,735 [26,951, 94,507]
PR3 bound	-	-	**48,730 [26,949, 94,497]**

Notice also that PR3 bound and PR3 Bernoulli value remaining are not evaluated in Scenarios 1 and 2 because their behavior is exactly the same as the Bernoulli value remaining and PR2 bound under Bernoulli distributions (Scenarios 1 and 2).

Looking at the average number of sample values and the confidence interval in Table 3, we find that the stopping criterion based on value remaining outperforms (i.e., requires a smaller number of samples than) the others for Scenarios 1 and 2, where the Bernstein criterion falls well behind the other criteria. In the third scenario, the performance for PR3 Bernoulli value remaining and PR3 bound is very similar, and they clearly outperform the other three stopping criteria. The Bernstein criterion is again quite far removed from the others. Thus, we discard this criterion for the Phase-2 simulation.

Note that although Adobe Target uses the empirical Bernstein criterion, we do not know how it is implemented so we cannot provide conclusions about the performance of Adobe Target stopping criteria.

5.2. Simulating A/B Testing

The empirical evaluation of MAB algorithms on real-world applications is complex and extremely expensive, mainly because we usually have no way of knowing the rewards associated with actions that have not been chosen, as mentioned in [5]. Evaluation involves executing the decision algorithms in real time in real environments and with real users across a time frame required to gain an accurate assessment of their convergence.

This then has to be repeated a considerable number of times to make statistically significant estimations. In this respect, [26] states "Compared to machine learning in the more standard supervised setting, evaluation of methods in a contextual bandit setting is frustratingly difficult... because of the interactive nature of the problem, it would seem that the only way to do this is to actually run the algorithm on 'live' data. However, in practice, this approach is likely to be infeasible due to the serious logistical challenges that it present".

To get an accurate perception of the requirements for evaluating this method in real environments, we have simulated 1,696,408,960 user interactions in the numerical test simulations in order to achieve statistical power for all the method and stopping criterion combinations within the four scenarios. Obviously it is only feasible to execute this number of interactions in a real environment for a very small subset of world companies.

Thus, we have to opt for one of the following alternatives:

- Simulate the evaluation environment such that the probability distribution of the arm rewards is as similar as possible to real environments. For delayed rewards, like A/B testing, we also have to simulate the probability distribution of the reward delay.
- Use a recommendation database, such as Yahoo Todays Module (https://webscope.sandbox.yahoo.com/catalog.php?datatype=r&did=49, accessed on 1 June 2021), and apply an offline evaluation method as in [5,26], replayed using previous random-scan data, to provide an unbiased estimator at the expense of considerably reducing the amount of effective data.

We decided to carry out a numerical simulation, since available offline databases do not totally fit our experiment requirements (they account for only binary rewards and not for feedback delay) mainly because off-line evaluations (e.g., [5,26]) are carried out only once.

Recommendation databases can be used to evaluate supervised algorithms, where the evaluation of each algorithm for each input datum depends not on previous evaluations but is, in itself, an experiment and is, consequently, sufficient for statistical significance. However, this does not apply with MAB algorithms, where each evaluation depends on the previous evaluations, where slightly different starting conditions could produce different convergence results. It is, therefore, necessary to repeat the experiment with different database subsets to achieve statistical significance.

Scenario Description

In the simulation process that we propose for the numerical analysis, we considered the most reliable environment possible, modeling the intrinsic properties of the systems architecture under consideration, as well as the behavior of customers accessing the systems.

First, customers enact a non-homogeneous Poisson process with the intensity function shown in Figure 1 to access the web. The average traffic is approximately 500 calls per day. This reflects how customers behave in digital environments. Most of these systems offer solutions for companies whose traffic volume is relatively low.

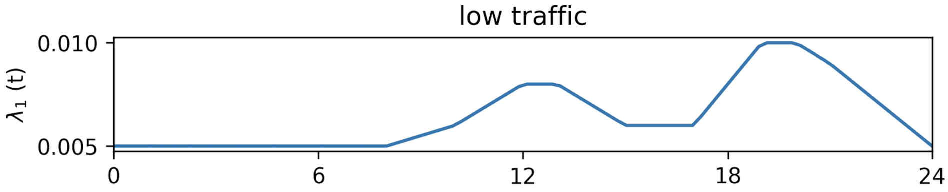

Figure 1. Intensity function for customer accesses.

Apart from the usual delays associated with the algorithm update (for example, probabilistic matching in Google, or Adobe Target), there is an intrinsic and inevitable delay from the moment the customer is offered a variation: he/she visualizes the web and decides to click or read the news. More customers may access the web during this time period, and the method will have to offer a variation without having updated the reward distribution. This delay will be simulated using an exponential distribution function with mean 150 s.

In our numerical analysis, we will consider four scenarios. The first two are based on the optimization of objectives that follow a Bernoulli distribution (success or failure). We assume that the performance of the above methods for this type of objective is very good and near optimum. In this case, we conduct a comparative analysis in terms of performance, and also evaluate whether PR2 bound improves the current state of the art for the proposed new stopping criteria, especially criteria to be applied in Bernoulli environments. Note that, in this context, PR3 bound is the same as PR2 bound, and the PR2 and PR3 value remaining methods are the same as the Bernoulli value remaining.

The following two scenarios are based on the optimization of objectives that do not follow a Bernoulli distribution. Here, we analyze whether or not the novelties proposed in this paper outperform the state of the art.

The *first scenario* is based on a context where the objective to be optimized is a conversion ratio (whether or not the customer clicks on a banner or button), comparing the current solution against one variation, which is the simplest case. We simulate a fairly common situation, also considered in [20], where the success rate is very low, 0.04, and the variation is slightly better, 0.05.

The *second scenario*, except that there are 10 variations, all with a low success rate. The best variation has a success value of 0.05, and the other variations return [0.045, 0.04, 0.04, 0.035, 0.035, 0.03, 0.03, 0.02, 0.02].

The first two scenarios are very common in online advertising contexts, where the best advertising option for a customer must be selected in the knowledge that they all have fairly low success rates.

The *third scenario* simulates a context where the customer is shown a short content, such as a news item or video. It is designed to maximize the time that the customer spends reading or paying attention to the content (when deciding whether or not to continue reading). In this case, the rewards are a continuous variable. As a very approximate simulation of real situations, a reward will be generated for each customer. This reward depends on whether or not the customer clicks on the news or the content. Then, a truncated exponential function models the time that the customer spends reading the news as a ratio or average time. In this scenario, the default news will be compared with other news. The parameters of the original and the variation are:

- Original: News item click ratio: 0.45; average and maximum reading time: 150 and 450 s, respectively.
- Variation: News item click ratio: 0.5; average and maximum reading time: 140 and 450 s, respectively.

The *fourth scenario* is similar to the third, albeit evaluating up to 9 new variations. The corresponding parameters are:

- Best variation: News item click ratio, 0.45; average and maximum reading time: 150 and 450 s, respectively.
- Original and other variations: News item click ratio, 0.5; average and maximum reading time: 140 and 450 s, respectively.

We have simulated these four scenarios because they are representative of real situations. In the first two scenarios, very common click conversions are used as the metric, and the performance of the proposed new methods is equal to Thompson sampling. In the third and fourth scenarios, on the other hand, the optimization metric depends on user resource consumption time, and the proposed new methods do not behave like Thompson sampling.

Finally, the MAB algorithms for dynamic traffic distribution tested in the scenarios under consideration are as follows:

- Thompson sampling (TS) for Bernoulli distributions of rewards (TS is the same as PR2 and PR3 in Bernoulli environments). This algorithm is used in Scenarios 1 and 2 only.
- Probabilistic matching (PM) with a short update window (window size = 1 h) and a long update window (window size = 12 h). These algorithms are used in Scenarios 1 and 2 only.
- ϵ-greedy, with $\epsilon = 0.2$.
- Classic A/B testing. The traffic is evenly distributed across all variations.
- PR2 and PR3, which are new methods proposed in this paper and are applicable only to Scenarios 3 and 4. As mentioned above, PR2 and PR3 are the same as TS in Scenarios 1 and 2.
- PR2 and PR3. As already mentioned, PR2 is the same as the purest version of Thomson sampling, i.e., for generic stochastic bandits (see [25]). Therefore, the results of PR2 will also refer to this algorithm.

We opted to analyze these methods rather than other methods because:

- The first three are TS variations for different update window sizes. The first is pure TS, where there is no window, the second has a short update window, and the third has a long update window. TS is one of the best-performing algorithms in environments with Bernoulli rewards, even in delay situations [13]. Additionally, many dynamic distribution products for A/B experiments appear to use different variations of TS (probabilistic matching) with different update window sizes (ABTasty, VWO, formerly Google Analytics). On this ground, we decided to use this method and two variations, increasing the update window size to analyze how they behave in the proposed scenarios.
- Even though it is a method with linear convergence, we analyze ϵ-greedy since it is very widespread in commercial products, such as Adobe Target, and we want to evaluate its behavior with respect to the state-of-the-art methods.

- Classic A/B testing is the reference method when traffic distribution is not dynamic. Therefore, it has to be measured for comparison with the default model.
- PR2 and PR3 are more recent MAB algorithms whose performance in similar contexts, as reported in [13], has been very good. We analyze their performance in A/B experimentation environments. As mentioned above, PR2 and PR3 are the same as TS in Scenarios 1 and 2. They are applicable only to Scenarios 3 and 4.
- We have not considered other MAB algorithms in the literature, such as UCB and its variations, because they appear to be at a clear disadvantage with respect to TS or TS for generic stochastic bandits [8,13].

The results of the simulations for the different scenarios are reported in the tables below. The tables include the accumulated mean regrets and corresponding standard deviation together with the mean number of samples in the simulation and the percentage error or false-positive rates. In addition, we calculated confidence intervals for both mean cumulative regrets and standard deviations using bootstrapping techniques and ranked all the results according to the upper percentile of the mean cumulative regret. To select the right statistical significance value α to bound the confidence intervals, we applied the Bonferroni correction to the desired level of $\alpha = 0.05$, returning confidence intervals of 99.5% for Scenarios 1 and 2 and 99.79% for Scenarios 3 and 4.

Besides, rigdeline plots illustrate the accumulated mean regret densities for the best four combinations of methods and stopping criteria, including a vertical red line reflecting the mean accumulated regret of the best method, and a blue line, corresponding to the regret derived by performing A/B testing using the classical test hypothesis.

5.3. Scenario 1 Results

Table 4 shows the Scenario 1 results for all the combinations of methods and stopping criteria under consideration. Mean values are provided in all columns derived from 500 simulations, and the methods are ranked from lowest to highest mean accumulated regret. 99.5% confidence intervals using the Bonferroni correction and bootstrapping methods have been added for mean accumulated regret and standard deviation values.

Table 4. Results for Scenario 1.

Method	Stopping Criteria	Accumulated Regret	Std Deviation	Samples	% Errors
PM_ws1	BerValRem	10.499 [8.884, 11.975]	13.504 [10.987, 15.957]	3432.00	7.6
TS	BerValRem	11.486 [9.903, 13.036]	14.422 [11.551, 17.126]	3873.30	8.0
PM_ws12	BerValRem	12.647 [10.949, 14.548]	14.614 [11.997, 17.155]	4262.70	4.8
A/B testing	BerValRem	13.433 [11.209, 15.775]	18.926 [14.753, 22.521]	2682.40	7.6
TS	PR2 bounds	23.941 [21.28, 26.472]	21.121 [18.469, 23.94]	24,565.28	0.8
PM_ws1	PR2 bounds	24.573 [21.564, 27.214]	23.217 [19.126, 28.836]	25,388.98	0.8
PM_ws12	PR2 bounds	24.668 [21.835, 27.221]	22.16 [18.353, 26.863]	27,184.72	0.2
e-greedy	BerValRem	25.575 [22.993, 27.971]	19.294 [16.142, 22.647]	5112.40	0.6
A/B testing	PR2 bounds	47.341 [42.44, 52.835]	38.406 [33.792, 42.531]	9463.96	0.2
e-greedy	PR2 bounds	53.958 [49.222, 58.875]	42.074 [37.183, 46.952]	10,792.18	1.0

First, we find that the methods based on probability matching (PM) and Thompson sampling (TS) with Bernoulli value remaining (BerValRem) as the stopping criterion provide the best results in terms of mean accumulated regrets. The results are better when the delay is small (PM_ws1) than for an online update like TS, possibly because more exploration is performed in the initial phases, which allows the algorithm to converge faster later. However, when the delay is greater, PM method performance drops (see, for example, the PM_ws12-configuration used by Google Analytics). Note again that TS is the same as PR2 and PR3 in Scenarios 1 and 2.

Besides, the algorithm with the worst performance is e-greedy based on the Adobe Target configuration parameters. The probable explanation for such poor performance, which is even worse than for A/B testing with equal traffic distribution, is the long warm-up period.

Note also that the performance of an algorithm based on uniformly distributed traffic with a dynamic stopping criterion based on Bernoulli value remaining (BerValRem) is not much worse than the optimal algorithm (ranking fourth in Table 4).

Besides, the accumulated regret densities for the combinations are shown in Figure 2. We find that the maximum accumulated regret for the PM_ws1 + BerValRem combination is again the lowest, followed by the TS + BerValRem and PM_ws2 + BerValRem combinations, which were ranked second and third, respectively, with respect to accumulated mean regret and the standard deviation, see Table 4.

Looking at the mean samples column in Table 4, we find that the stopping criteria values for combinations with Bernoulli value remaining (BerValRem) are quite a lot lower than for PR2 bounds. Thus, the new stopping criteria proposed in this paper do not result in any improvement in this scenario. A/B testing + BerValRem is the combination that requires a smaller mean number of samples (2682.4), followed by PM_ws1 + BerValRem, TS + BerValRem and PM_ws2 + BerValRem.

However, we find that the percentage error or false-positive rate are between 4.8 and 7.6. This can be considered significantly high. This is a drawback of these combinations.

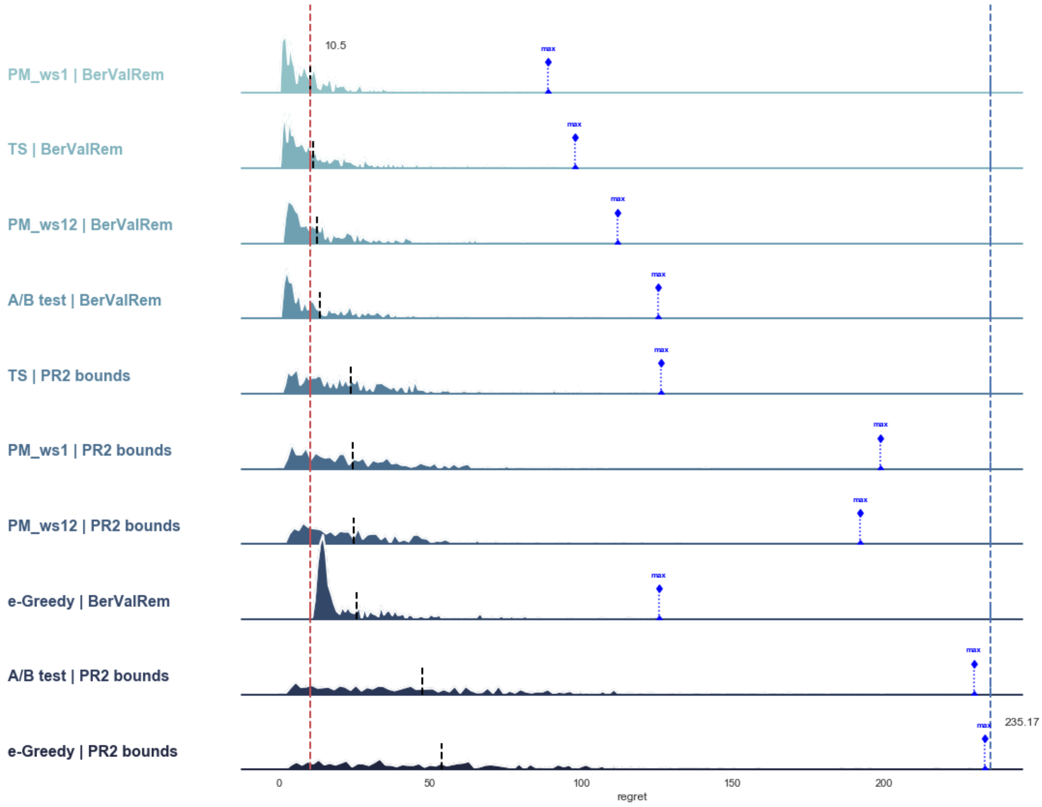

Figure 2. Ridgeline plots for Scenario 1.

5.4. Scenario 2 Results

Table 5 shows the Scenario 2 results for all the combinations of methods and stopping criteria under consideration. Mean values are provided in all columns derived from 500 simulations, and the methods are listed from lowest to highest mean accumulated regret. 99.5% confidence intervals using the Bonferroni correction and bootstrapping methods have been added for mean accumulated regret and standard deviation values.

Table 5. Results for Scenario 2.

Method	Stopping Criteria	Accumulated Regret	Std Deviation	Samples	% Errors
PM_ws1	BerValRem	194.555 [185.843, 201.895]	73.961 [63.911, 82.649]	60,639.18	0.0
PM_ws12	BerValRem	194.785 [185.121, 205.635]	77.099 [64.384, 93.904]	61,127.26	0.6
TS	BerValRem	197.545 [187.344, 207.089]	83.44, [69.08, 99.238]	63,312.14	0.4
PM_ws1	PR2 bounds	218.134 [208.681, 226.015]	80.993, [72.323, 91.895]	105,112.06	0.0
PM_ws12	PR2 bounds	220.427 [211.444, 232.338]	94.451 [67.116, 128.154]	105,706.12	0.4
TS	PR2 bounds	220.561 [211.506, 232.514]	89.888 [75.233, 106.156]	105,939.20	0.4
e-greedy	BerValRem	1208.136 [1124.058, 1302.655]	623.357, [541.329, 695.437]	87,840.12	0.2
e-greedy	PR2 bounds	1732.984 [1610.772, 1862.06]	1029.226 [919.671, 1122.726]	128,500.10	0.2
A/B testing	BerValRem	1960.025 [1813.084, 2106.305]	1329.872 [1139.229, 1519.782]	126,450.30	0.2
A/B testing	PR2 bounds	3324.493 [3003.617, 3698.993]	2402.718 [2177.281, 2648.417]	214,477.92	0.0

As in Scenario 1, the methods based on probability matching (PM) and TS with the Bernoulli value remaining (BerValRem) stopping criterion are the best in terms of mean accumulated regrets. In this case, the performance of the first three methods is significantly similar. Moreover, the three best-ranked combinations in terms of mean accumulated regrets are also the best three combinations (in the same order) with respect to the standard deviations (dispersion of accumulated regrets), the maximum accumulated regret (see Figure 3) and the number of mean samples required in the simulations.

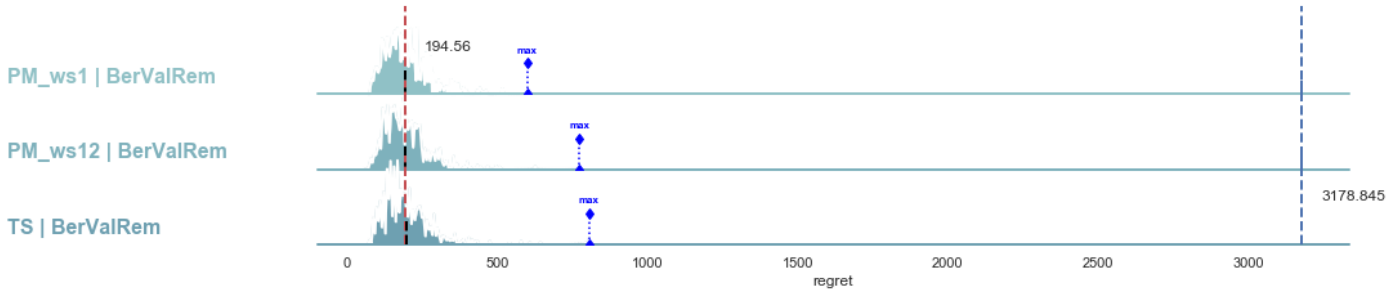

Figure 3. Ridgeline plots for Scenario 2.

Now, the uniformly distributed traffic (A/B testing) methods are clearly outperformed by the others, including e-greedy, see Table 5. Thus, the classical methods based on uniformly distributed traffic (A/B testing) perform worse, as the number of samples required to achieve experiment statistical significance increases, even with a dynamic stopping criteria.

As in Scenario 1, the methods whose stopping criterion is based on PR2 bound are clearly worse than others based on Bernoulli value remaining (BerValRem). Therefore, we conclude that they do not constitute an improvement with respect to the state of the art in environments with Bernoulli rewards.

Finally, the percentage error or false-positive rate values are low for all combinations $(0.2, 0.6)$, where PM_ws1 + BerValRem was again the best combination, scoring 0.

5.5. Scenario 3 Results

Table 6 shows the Scenario 3 results for all the method combination and stopping criteria under consideration. 99.79% confidence intervals using the Bonferroni correction and bootstrapping methods have been added for mean accumulated regret and standard deviation values.

Table 6. Results for Scenario 3.

Method	Stopping Criteria	Accumulated Regret	Std Deviation	Samples	% Errors
PR3	PR3 ValRem	17.509 [15.827, 19.853]	16.144 [13.699, 18.888]	4769.70	1.2
PM_ws1	PR3 ValRem	18.05 [16.127, 20.453]	16.982 [14.347, 19.926]	4259.90	1.6
PR2	PR3 ValRem	19.227 [16.933, 21.271]	17.641 [14.993, 20.584]	4502.20	1.0
PR3	PR2 ValRem	19.876 [17.61, 22.115]	18.202 [15.49, 20.95]	6904.60	1.2
PM_ws12	PR3 ValRem	20.527 [18.433, 22.688]	17.217 [15.126, 19.919]	4838.10	0.8
PM_ws1	PR2 ValRem	20.582 [18.333, 23.612]	19.577 [16.725, 22.604]	5568.30	1.4
PR2	PR2 ValRem	22.136 [19.675, 24.675]	19.576 [17.278, 22.308]	6081.30	0.8
PM_ws12	PR2 ValRem	23.304 [21.076, 25.76]	19.576 [17.278, 22.308]	6363.70	0.4
A/B testing	PR3 ValRem	24.874 [22.261, 28.047]	21.267 [18.116, 24.911]	3302.70	1.0
PR3	PR3 bounds	27.955 [25.576, 30.217]	18.708 [16.434, 20.804]	19,051.18	0.0
e-greedy	PR3 ValRem	29.852 [27.74, 32.496]	18.104 [14.811, 21.628]	3964.10	0.0
A/B testing	PR2 ValRem	31.11 [27.457, 34.69]	27.528 [23.505, 31.108]	4132.00	0.8
PR2	PR3 bounds	31.754 [29.099, 34.976]	23.472 [18.746, 28.531]	14,605.28	0.0
PM_ws1	PR3 bounds	32.38 [29.734, 35.136]	22.2 [19.491, 25.2]	15,176.76	0.0
PM_ws12	PR3 bounds	32.816 [29.677, 35.753]	21.22 [18.407, 23.826]	15,617.36	0.0
PR3	PR2 bounds	35.214 [32.089, 37.893]	21.831 [19.187, 24.287]	57,131.10	0.0
e-greedy	PR2 ValRem	36.068 [32.474, 39.552]	23.865 [19.76, 28.499]	4786.40	0.0
PR2	PR2 bounds	40.967 [37.335, 44.977]	26.863 [22.931, 31.962]	34,122.30	0.0
PM_ws12	PR2 bounds	41.784 [38.635, 44.984]	25.111 [22.66, 27.832]	35,956.58	0.0
PM_ws1	PR2 bounds	41.818 [38.746, 45.56]	26.908 [23.676, 30.295]	35,676.58	0.0
A/B testing	PR3 bounds	57.497 [53.691, 62.432]	33.763 [30.165, 37.272]	7638.42	0.0
e-greedy	PR3 bounds	58.759 [54.506, 62.56]	32.715 [28.831, 36.662]	7801.54	0.0
A/B testing	PR2 bounds	95.405 [88.663, 102.207]	50.954 [46.493, 55.533]	12,673.12	0.0
e-greedy	PR2 bounds	96.279 [90.85, 102.754]	50.682 [46.421, 56.148]	12,786.00	0.0

In this scenario, PR3 + PR3 ValRem is the best combination, followed by PM_ws1 + PR3 ValRem and PR2 + PR3 ValRem in terms of mean accumulated regrets. Moreover, the three best-ranked combinations in terms of mean accumulated regrets are also the best three combinations (in the same order) with respect to standard deviations (dispersion of accumulated regrets). However, PM_ws1 + PR3 ValRem slightly outperforms the other two combinations in terms of maximum accumulated regret (see Figure 4).

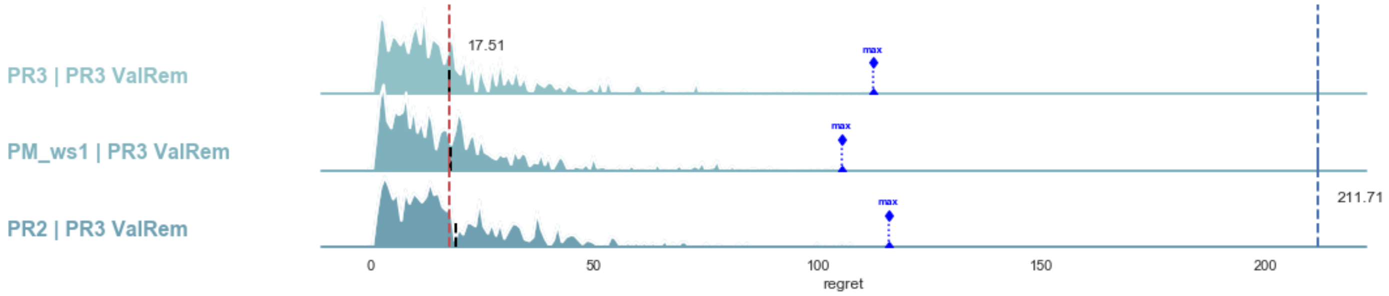

Figure 4. Ridgeline plots for Scenario 3.

As in Scenario 1, the e-greedy method output the worst results due to its warm-up phase.

Regarding the stopping criteria, we find, looking at the mean samples column, that the values for combinations with the PR3 value remaining (PR3 ValRem) stopping criterion are the lowest, followed by PR2 ValRem and PR2 and PR3 bounds. Note that the PR2 and PR3 ValRem stopping criteria are the original contributions in this paper. The three best-ranked combinations in terms of mean accumulated regrets are among the best with respect to the number of mean required samples.

Finally, percentage error or false-positive rate values are low for all combinations $(0, 1.6)$, with the combination PR3 + PR3 ValRem combination scoring 1.2.

5.6. Scenario 4 Results

Table 7 shows the Scenario 4 results for all method combinations and stopping criteria under consideration. 99.79% confidence intervals using the Bonferroni correction and bootstrapping methods have been added for mean accumulated regret and standard deviation values.

In this scenario, PR3 is again the best method in terms of mean accumulated regrets, combined with PR3 bounds as the stopping criterion. For all methods, the mean accumulated regrets are lower when the PR3 bound stopping criterion is used, followed by the PR3 RemVal, and PR2-based stopping criteria. However, the standard deviations (dispersion of accumulated regrets) and maximum accumulated regret (see Figure 5) are slightly better when using PR3 RemVal than PR3 bounds for all methods.

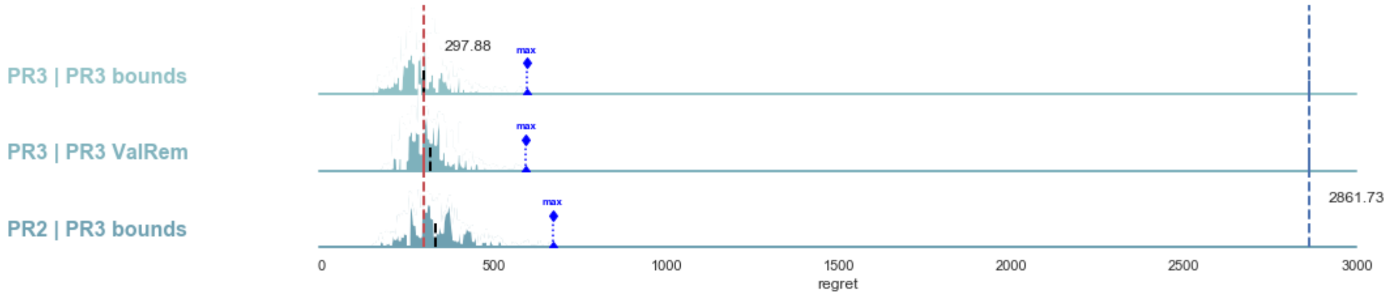

Figure 5. Ridgeline plots for Scenario 4.

Table 7. Results for Scenario 4.

Method	Stopping Criteria	Accumulated Regret	Std Deviation	Samples	% Errors
PR3	PR3 bounds	297.885 [288.683, 307.784]	72.661 [64.164, 79.885]	58,374.62	0.0
PR3	PR3 ValRem	314.402 [306.081, 322.783]	65.726 [57.646, 73.918]	71,610.50	0.0
PR2	PR3 bounds	331.952 [320.883, 341.7]	84.417 [76.892, 92.593]	47,568.54	0.0
PM_ws1	PR3 bounds	336.177 [324.022, 349.837]	98.973 [88.272, 112.498]	48,150.88	0.0
PR2	PR3 ValRem	342.995 [332.6, 353.55]	77.049 [68.029, 86.996]	51,962.66	0.0
PM_ws12	PR3 bounds	344.552 [331.648, 357.038]	92.832 [83.426, 106.653]	49,327.28	0.0
PM_ws1	PR3 ValRem	344.185 [331.093, 358.874]	98.207 [85.103, 110.513]	51,565.74	0.0
PM_ws12	PR3 ValRem	349.415 [339.469, 362.155]	88.82 [77.917, 100.732]	51,626.04	0.0
PR3	PR2 bounds	378.907 [370.474, 387.263]	71.807 [64.846, 80.103]	169,198.04	0.0
PR3	PR2 ValRem	395.031 [385.282, 403.973]	71.38 [64.317, 78.215]	207,593.72	0.0
PM_ws1	PR2 ValRem	419.929 [406.494, 430.834]	97.844 [85.994, 109.369]	87,988.90	0.0
PR2	PR2 ValRem	423.064 [410.97, 434.289]	82.953 [75.133, 90.271]	91,690.60	0.0
PM_ws12	PR2 ValRem	428.112 [416.32, 440.194]	96.57 [86.796, 107.049]	88,687.40	0.0
PR2	PR2 bounds	433.309 [421.156, 445.92]	88.816 [81.052, 96.499]	99,834.48	0.0
PM_ws1	PR2 bounds	436.013 [424.017, 448.782]	100.925 [90.395, 111.934]	99,408.86	0.0
PM_ws12	PR2 bounds	445.884 [433.262, 459.936]	100.957 [90.125, 111.406]	101,205.00	0.0
e-greedy	PR3 ValRem	531.664 [510.642, 555.046]	185.89 [168.448, 201.686]	41,856.16	0.0
e-greedy	PR3 bounds	627.735 [603.311, 653.955]	209.193 [190.073, 231.37]	49,635.12	0.0
A/B testing	PR3 ValRem	683.777 [645.122, 718.451]	285.261 [256.161, 318.201]	50,443.04	0.0
e-greedy	PR2 ValRem	725.434 [691.615, 758.818]	241.654 [217.532, 268.492]	57,547.42	0.0
A/B testing	PR3 bounds	913.549 [868.428, 953.342]	359.637 [315.555, 401.617]	67,394.40	0.0
e-greedy	PR2 bounds	957.14 [918.402, 990.533]	281.442 [254.412, 307.287]	76,309.54	0.0
A/B testing	PR2 ValRem	947.351 [905.521, 992.398]	357.559 [316.945, 392.403]	69,888.32	0.0
A/B testing	PR2 bounds	1407.125 [1336.966, 1467.677]	470.799 [421.703, 520.804]	103,809.12	0.0

As in Scenario 2, the method with uniformly distributed traffic (A/B testing) is again the worst performer.

Regarding the stopping criteria, PR3 bounds is used in four out of the five best combinations in terms of mean samples, but the differences between PR3 bounds and PR3 RemVal are generally small.

Finally, percentage error or false-positive rate values are 0 for all the combinations under analysis.

6. Conclusions

The first thing to be highlighted is that none of the method + stopping criterion combinations outperforms the other combinations in all the four scenarios under consideration. Besides, it is clear that the use of the homogeneous traffic distribution method (denoted as A/B testing in the simulation process) already constitutes a considerable improvement over methods based on classical testing hypotheses.

In the scenarios where the rewards follow a Bernoulli distribution (Scenarios 1 and 2), it is perfectly documented that vendors use dynamic traffic distribution by means of multi-armed bandit (MAB) methods. The best solution is a Bayesian approach with probabilistic matching or Thompson sampling. Moreover, methods that update weights at small, for example, one-hour intervals (PM_ws1), like Optimizely and ABTasty, even appear to have a slight performance edge over online updating methods. However, performance appears to drop when the update window is larger, for example using Google Analytics with a 12-hour window (PM_ws12).

Bernoulli value remaining (BerValRem) is clearly by far the most efficient stopping criterion for Bernoulli distributed scenarios, although there is some doubt, above all in the first case, as to percentage error, where the resulting values are greater than those specified in the design (1%). Besides, it is also evident that the PR2 bounds stopping criteria, proposed in this paper in no case improves the state of the art for Bernoulli environments.

Regarding the unknown non-Bernoulli distribution Scenarios (3 and 4), i.e., scenarios where the methods proposed in this paper really apply, PR3 outperforms all the state-of-the-art methods. Except for Adobe Target, where it is unclear which MAB algorithm and stopping criterion is used or whether or not it accounts for dynamic traffic distribution, none of the vendors clearly describe how they implement their solutions in non-Bernoulli environments. In any case, the simulations were carried out assuming that the vendors used the best methods known in the literature in these situations: Thompson sampling for general stochastic bandits in combination with one of the stopping criteria considered in this paper.

In Scenarios 3 and 4, the most efficient stopping criteria are PR3 value remaining (PR3 ValRem) and PR3 bounds. PR3 value remaining is the best when not many experiments are required to achieve statistical significance, and PR3 bounds when the number of experiments increases (due to a greater number of variations or a smaller difference between the best and the other variations). However, PR3 value remaining requires more computing resources than PR3 bounds, since it entails simulation. In production environments, a finer adjustment can be made by combining the two methods, where PR3 bounds is used merely to check if a stopping threshold has been reached, and then PR3 value remaining is used for the latest experiments or iterations.

We should also emphasize that the PR3 algorithm is equivalent and behaves exactly like Thompson Sampling in Bernoulli rewards environments, also offering optimal performance. Therefore, we can conclude that it is not necessary to use one algorithm or another according to the type of objective to be measured (scenario) in the experiment, PR3 is a good option in all cases where the stopping criterion should be value remaining for Bernoulli rewards environments, and PR3 bounds for non-Bernoulli environments.

Last but not least, according to the cost (in terms of number of samples) required to achieve significant scientific conclusions about executed experiments (based on null hypothesis rejection), all the combinations offer a substantial improvement on classical techniques where the number of samples for each variation have to be established a priori or under conditions where traffic allocation is not dynamic (denoted as A/B testing in the simulation process).

Prospective future lines of research should address more complex simulation environments that take into account other metrics than described in the four scenarios (e.g., valuation ratings or sale prices), as well as evaluating the best alternatives in real production environments. It would also be interesting to implement this type of multivariate dynamic traffic distribution and stopping criteria using, for example, contextual MAB techniques.

Author Contributions: Conceptualization, M.M, A.J.-M., A.M. and J.Z.H.; methodology, M.M., A.J.-M., A.M. and J.Z.H.; software, M.M.; validation, M.M., A.J.-M., A.M. and J.Z.H.; formal analysis, M.M., A.J.-M., A.M. and J.Z.H.; resources, M.M.; writing—original draft preparation, M.M. and A.J.-M.; writing—review and editing, M.M. and A.J.-M.; project administration, A.J.-M. and A.M.; funding acquisition, A.J.-M. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Spanish Ministry of Economy and Competitiveness project grant number MTM2017-86875-C3-3-R.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code created to carry out the simulation is available at https://github.com/mmartinb75/pyBanditsWithForTestAB (accessed on 1 June 2021), whereas all the data generated by the numerical analysis is available at https://drive.google.com/drive/folders/14dUeuG_bPHfdoaQpTu8jvHLKiMyHSSFP?usp=sharing, accessed on 1 June 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Google. Google Analytics Help. Available online: https://analytics.googleblog.com/2013/01/multi-armed-bandit-experiments.html (accessed on 23 July 2021).
2. Optimizely. Available online: https://www.optimizely.com/ (accessed on 23 July 2021).
3. Audibert, J.Y.; Bubeck, S. Regret bounds and minimax policies under partial monitoring. *J. Mach. Learn. Res.* **2010**, *11*, 2785–2836.
4. Baransi, A.; Maillard, O.A.; Mannor, S. Sub-sampling for multi-armed bandits. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Nancy, France, 14–18 September 2014; 13p.
5. Chapelle, O.; Li, L. An empirical evaluation of Thompson sampling. *Adv. Neural Inf. Process. Syst.* **2011**, *17*, 2249–2257.
6. Garivier, A.; Cappé, O. The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond. *J. Mach. Learn. Res. Workshop Conf. Proc.* **2011**, *19*, 359–376.
7. Kaufmann, E.; Cappé, O.; Garivier, A. On Bayesian upper confidence bounds for bandit problems. In Proceedings of the Artificial Intelligence and Statistics Conference, La Palma, Spain, 21–23 April 2012; pp. 592–600.
8. Martín, M.; Jiménez-Martín, A.; Mateos, A. Possibilistic Reward Method for the Multi-Armed Bandit Problem. *Neurocomputing* **2018**, *310*, 201–212. [CrossRef]
9. Ju, N.; Hu, D.; Henderson, A.; Hong, L. A sequential test for selecting the better variant, Online A/B testing, adaptive allocation, and continuous monitoring. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 February 2019; pp. 492–500.
10. Mattos, D.I.; Jan, B.; Olsson, H.H. Multi-armed bandits in the wild: Pitfalls and strategies in online experiments. *Inf. Softw. Technol.* **2019**, *113*, 68–81. [CrossRef]
11. Johari, R.; Walsh, D.; Pekelis, L. Peeking at A/B Tests: Why it matters, and what to do about it. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 1517–1525.
12. Johari, R.; Koomen, P.; Pekelis, L.; Walsh, D. Always Valid Inference: Continuous Monitoring of A/B Tests. *Oper. Res.* **2021**. [CrossRef]
13. Martín, M.; Jiménez-Martín, A.; Mateos, A. A Numerical Analysis of Allocation Strategies for the Multi Armed Bandit Problem under Delayed Rewards Conditions in Digital Campaign Management. *Neurocomputing* **2019**, *363*, 99–113. [CrossRef]
14. Martín, M.; Jiménez-Martín, A.; Mateos, A. The Possibilistic Reward Method and a Dynamic Extension for the Multi-armed Bandit Problem: A Numerical Study. In Proceedings of the 6th International Conference on Operations Research and Enterprise Systems, Porto, Portugal, 23–25 February 2017; pp. 75–84.
15. VWO. Multi-Armed Bandit (MAB)—A/B Testing Sans Regret. Available online: https://vwo.com/blog/multi-armed-bandit-algorithm/ (accessed on 23 July 2021).
16. ABTasty. Do Not Lose Out on Conversions During Your A/B Tests. Available online: https://www.abtasty.com/blog/dynamic-traffic-allocation/ (accessed on 23 July 2021).

17. Adobe. Adobe Target Automatic Traffic Allocation. Available online: https://docs.adobe.com/content/help/en/target/using/activities/auto-allocate/automated-traffic-allocation.html (accessed on 23 July 2021).
18. Martín, M.; Jiménez-Martín, A.; Mateos, A. A/B Testing Adaptations based on Possibilistic Reward Methods for Checkout Processes: A Numerical Analysis. In Proceedings of the 9th International Conference on Operations Research and Enterprise Systems, Valleta, Malta, 22–24 February 2020; pp. 278–285.
19. Dunn, O.J. Multiple comparisons among means. *J. Am. Stat. Assoc.* **1961**, *56*, 52–64. [CrossRef]
20. Scott, S.L. Multi-armed bandit experiments in the online service economy. *Appl. Stoch. Model. Bus. Ind.* **2015**, *31*, 37–45. [CrossRef]
21. Bernstein, S.N. *Probability Theory*; GTTI: Sutton West, ON, Canada, 1946.
22. Audibert, J.-Y.; Munos, R.; Szepesvári, C. Tuning bandit algorithms in stochastic environments. In Proceedings of the 18th International Conference, ALT 2007, Sendai, Japan, 1–4 October 2007; pp. 150–165.
23. Hoeffding, W. Probability inequalities for sums of bounded random variables. *Adv. Appl. Math.* **1963**, *58*, 13–30. [CrossRef]
24. Smets, P. Data Fusion in the Transferable Belief Model. In Proceedings of the Third International Conference on Information Fusion, Paris, France, 10–13 July 2000; pp. 21–33.
25. Agrawal, S.; Goyal, N. Analysis of Thompson Sampling for the multi-armed bandit problem. *Conf. Learn. Theory* **2012**, *39*, 1–26.
26. Lihong, L.; Langford, J.; Schapire, R.E. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 661–670.