(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 August 2023 (17.08.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/154337 A1

(51) International Patent Classification:
G06Q 30/0203 (2023.01) *H04L 67/53* (2022.01)
G06F 11/36 (2006.01) *G06F 11/34* (2006.01)
G06Q 30/0242 (2023.01)

(21) International Application Number:
PCT/US2023/012611

(22) International Filing Date:
08 February 2023 (08.02.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/308,700 10 February 2022 (10.02.2022) US
17/887,195 12 August 2022 (12.08.2022) US

(71) Applicant: **HOME DEPOT INTERNATIONAL, INC.** [US/US]; 2455 Paces Ferry Road, Atlanta, Georgia 30339 (US).

(72) Inventors: **XIANG, Ding**; c/o Home Depot International, Inc., 2455 Paces Ferry Road, Atlanta, Georgia 30339 (US). **WANG, Jiaqi**; c/o Home Depot International, Inc., 2455 Paces Ferry Road, Atlanta, Georgia 30339 (US). **WEST, Rebecca**; c/o Home Depot International, Inc., 2455 Paces Ferry Road, Atlanta, Georgia 30339 (US). **CUI, Xiquan**; c/o Home Depot International, Inc., 2455 Paces Ferry Road, Atlanta, Georgia 30339 (US). **HUANG, Jinzhou**; c/o Home Depot International, Inc., 2455 Paces Ferry Road, Atlanta, Georgia 30339 (US).

(74) Agent: **GIROUX, Jonathan** et al.; Greenberg Traurig, LLP, 77 W. Wacker Drive, Suite 3100, Chicago, Illinois 60601 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,

(54) Title: INFORMATION-GREEDY MULTI-ARM BANDITS FOR ELECTRONIC USER INTERFACE EXPERIENCE TESTING

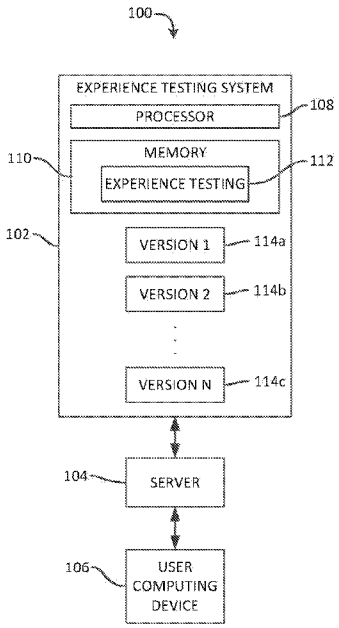

FIG. 1

(57) Abstract: A method for determining a user experience for an electronic user interface includes defining a test period for testing two or more versions of an electronic user interface, receiving, from each of a plurality of users during the test period, a respective request for the electronic user interface, determining, for each of the plurality of users, a respective version of the two or more versions of the electronic user interface by maximizing test power during the test period while maintaining higher in-test rewards than an A/B test or maximizing the rewards during the test period while maintaining a test power no worse than an A/B test, and causing, for each of the plurality of users, the determined version of the electronic user interface to be delivered to the user.

WO 2023/154337 A1

[Continued on next page]

DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

INFORMATION-GREEDY MULTI-ARM BANDITS FOR ELECTRONIC USER INTERFACE EXPERIENCE TESTING

Field of the Disclosure

[0001] The present disclosure generally relates to website experience testing, including multi-arm bandit methods for website experience testing.

Brief Description of the Drawings

[0002] FIG. 1 is a block diagram illustrating an example system for deploying a website experience testing algorithm.

[0003] FIG. 2 is a flow chart illustrating an example method of delivering a respective website experience to each of a plurality of users.

[0004] FIG. 3 is a table illustrating results of tests performed according to the novel approaches of this disclosure compared to known approaches.

[0005] FIG. 4 is a table illustrating results of tests performed according to the novel approaches of this disclosure compared to known approaches.

[0006] FIG. 5 is a plot illustrating results of tests performed according to the novel approaches of this disclosure compared to known approaches.

[0007] FIG. 6 is a plot illustrating results of tests performed according to the novel approaches of this disclosure compared to known approaches.

[0008] FIG. 7 is a series of bar graphs illustrating results of tests performed according to the novel approaches of this disclosure compared to known approaches.

[0009] FIG. 8 is a series of plots illustrating results of tests performed according to the novel approaches of this disclosure compared to known approaches.

[0010] FIG. 9 is a series of plots illustrating results of tests performed according to the novel approaches of this disclosure compared to known approaches.

[0011] FIG. 10 is a block diagram view of a user computing environment.

Detailed Description

[0012] Current approaches for testing different website experiences either do not appropriately maximize the power of the test, or do not maximize the rewards associated with the testing period. For example, A/B tests generally assign users to the experiences under test at random and with equal probability. As a result, a typical A/B test does not maximize the rewards of the testing period, particularly when one of the experiences under test clearly underperforms. In another example, a typical multi-arm bandit (MAB) approach may maximize in-test rewards, but has relatively low testing power because different experiences are tested in different quantities. An experience testing approach according to the present

disclosure may maximize both test power and in-test rewards, improving upon both A/B testing and typical MAB approaches.

[0013] Referring to the drawings, wherein like numerals refer to the same or similar features in the various views, FIG. 1 is a block diagram illustrating an example system 100 for performing a an experience test for an electronic user interface, such as a website or a mobile device application, and deploying a most successful tested experience. The system 100 may include an experience testing system 102, a server 104, and a user computing device 106.

[0014] The experience testing system 102 may include a processor 108 and a non-transitory, computer readable memory 110 storing instructions that, when executed by the processor 108, cause the system 102 to perform one or more processes, methods, algorithms, steps, etc. of this disclosure. For example, the memory may include an experience testing module 112 configured to conduct a test of a plurality of website experiences.

[0015] The experience testing system 102 may be deployed in connection with an electronic user interface, such as a website or mobile application hosted by the server 104 for access by the user computing device 106 and/or a plurality of other user computing devices. The experience testing system may test different experiences on the interface to determine a preferred experience going forward. Experiences may include, for example, different layouts of the interface, different search engine parameter settings, different document recommendation strategies, and/or any other setting or configuration of the interface that may affect the user experience on the interface. For example, experiences may be represented in various versions 114a, 114b, 114c of a portion of the website or other electronic user interface (which may be referred to herein individually as a version 114 or collectively as the versions 114).

[0016] To conduct an experience test, the experience testing system 102 may cause the server 104 to provide one of the different experiences (e.g., one of versions 114a, 114b, 114c) to each of a plurality of different users according to a particular strategy. For example, in a traditional A/B test strategy, the server 104 would provide a randomly-selected one of the experiences to each user, with each experience having an equal probability of being provided by the server 104. Two particular novel approaches, which may be referred to as information-greedy MAB approaches, are described herein.

[0017] In a first example, an information-greedy multi-arm bandit may seek to maximize test power during the test period while maintaining higher in-test rewards than an A/B test. For example, in some embodiments, an information-greedy MAB deployed to test two

experiences may calculate a ratio of the total number of times each experience has been provided to users, calculate a square root of a ratio of experience cumulative rewards, where the cumulative rewards for each experience are calculated as a product of the cumulative reward of the experience and one-minus that reward (where a reward is expressed as a zero or one), and compare the number of times ratio to the square root to determine the appropriate experience to serve.

[0018] In a second example, an information-reward-greedy MAB may also seek to maximize the rewards during the test period while maintaining a test power no worse than an A/B test. For example, in some embodiments, an information-reward-greedy MAB deployed to test two experiences may calculate a ratio of the total number of times each experience has been provided to users, calculate a ratio of experience cumulative rewards, where the cumulative rewards for each experience are calculated as a product of the cumulative reward of the experience and one-minus that reward (where a reward is expressed as a zero or one), and compare the two ratios to determine the appropriate experience to serve.

[0019] The experience testing module 112 may conduct a test of the various versions 114 during a predefined test period in order to determine a preferred one of the versions 114. The predefined test period may be or may include a predefined number of test users, in some embodiments. Additionally or alternatively, the predefined test period may be or may include a time period. After the test period, the preferred version (e.g., version 114b) may be provided to users going forward.

[0020] Although experience testing is described herein as being performed by a backend system associated with a server, it should be understood that some or all aspects of an experience test may instead be performed locally (e.g., on one or more user computing devices 106). For example, the functionality of the experience testing system 102 may be implemented on a user computing device 106. For example, a user computing device 106 may have the versions 114 stored on the memory of the user computing device 106, and the user computing device 106 may determine rewards associated with a given instance of a particular version being selected, and may report that reward back to a backend system that performs version selection according to rewards determined by many user computing devices 106 according to their respective experiences.

[0021] FIG. 2 is a flow chart illustrating an example method 200 of determining a user experience for an electronic user interface. The method 200, or one or more portions of the method 200, may be performed by the experience testing system 102, in some embodiments.

[0022] The method 200 may include, at block 202, defining a test period for testing two or more versions of an electronic user interface. The two or more versions may be or may include different user experiences on the electronic user interface. The electronic user interface may be or may include a website, a webpage, or a portion of a webpage, for example. The different user experiences may be or may include different search engine parameter settings, different document recommendation strategies, and/or any other setting or configuration of the interface that may affect the user experience on the interface. The test period may be defined to include a time period, a quantity of tested users, a quantity of tests of one or more of the versions (e.g., all versions), and/or some other parameter. Additionally or alternatively, the test period may be defined as a period necessary to determine a superior version of the interface by a minimum threshold, as discussed below.

[0023] In some embodiments, block 202 may additionally include defining the two or more versions. In some embodiments, the versions may be defined automatically (e.g., through algorithmic or randomized determination of a page configuration, algorithmic determination of a set of search engine parameter values, etc. In some embodiments, the versions may be defined manually.

[0024] The method 200 may further include, at block 204, receiving, from each of a plurality of first users during the test period, a respective request for the electronic user interface. Each user request may be, for example, a request for (e.g., attempt to navigate to) a portion of the electronic user interface that includes the relevant version. For example, where the difference between versions is different search engine parameter settings, requests received at block 204 may include search requests by the users. In another example, where the difference between versions is different layout for a home page of a website, requests received at block 204 may include user requests for the website domain through their browser.

[0025] The method may further include, at block 206, determining, for each of the plurality of first users, a respective version of the two or more versions of the electronic user interface. The determination at block 206 may include maximizing test power during the test period while maintaining higher in-test rewards than an A/B test or maximizing the rewards during the test period while maintaining a test power no worse than an A/B test. Both of these options are referred to herein as "information greedy MAB" approaches. These two approaches are discussed in turn below.

[0026] Before describing the information greedy MAB algorithms in detail, the general model formulation and notations for MAB algorithms is first described. Assume we have K

competing versions or experiences (also known as "arms" of a MAB test), denoted by set $E = \{1, 2, ..., K\}$, and a decision strategy $\boldsymbol{S}$ such that for every customer's visit at time t, the strategy $\boldsymbol{S}$ can decide which one of the experiences, $e_t \in E$, to show. After showing the experience e_t, we will see a feedback or reward, denoted by r_t, from the user who received the experience. The feedback could either be Boolean or binary ($r_t \in \{0, 1\}$) such as the experience being click or not, and a purchase being made or not, or continuous ($r_t \in \mathbb{R}, r_t \geq 0$) such as the total price of the order, and the dwelling time on that experience etc. For this work, we focus on the binary feedback or reward, i.e., $r_t \in \{0, 1\}$ and $r_t = 1$ meaning positive feedback such as an effective purchase, click, etc., and $r_t = 0$ meaning negative feedback (or no feedback), such as by the user not performing any desired action after being delivered the selected interface version.

[0027] Assume the probability of getting a reward of 1 for showing an experience $e \in E$ is $p(e)$, and it is unchanged overtime. Assume users visit in a time sequence $(t_1, t_2, t_3, ...)$ denoted by $(t_i)_{i=1}^{\infty}$, where $t_1 \leq t_2 \leq t_3 \leq ...$, which allows multiple visits at the same time. Also, the superscript ∞ can be replaced by a finite number if only a fixed time range is considered; this is also true for the sequences below). At each visit, a version is decided and delivered to a user by using strategy $\boldsymbol{S}$. Then we have a logging of the delivered experiences and corresponding rewards, i.e., a sequence of experience-reward pairs $((e_{t_1}, r_{t_1}), (e_{t_2}, r_{t_2}), (e_{t_3}, r_{t_3}), ...)$, denoted by $(e_{t_i}, r_{t_i})_{i=1}^{\infty}$.

[0028] At any time t_n, the performance of an experience e in E may be measured, using the logging generated by strategy $\boldsymbol{S}$ up to time t_n, $(e_{t_i}, r_{t_i})_{i=1}^{n}$. Equation 1 below describes the total number of times $N_{t_n}(e)$ that a version e has been provided to users through time t_n:

$$N_{t_n}(e) = \sum_{i=1}^{n} \mathbb{1}(e = e_{t_i}), for\ e \in E \quad \text{(Eq. 1)}$$

where $\mathbb{1}(\cdot)$ is the indicator function. The collective total number n of times that all versions have been provided to users through time t_n is shown in equation 2 below:

$$n = \sum_{e \in E} N_{t_n}(e) \quad \text{(Eq. 2)}$$

[0029] The total reward R for showing version e up to time t_n is shown in equation 3 below:

$$R_{t_n}(e) = \sum_{i=1}^{n} r_{t_i} \mathbb{1}(e = e_{t_i}), for\ e \in E \quad \text{(Eq. 3)}$$

[0030] The average reward p for showing experience e up to time t_n, is shown in equation 4 below:

$$p_{t_n}(e) = \frac{R_{t_n}(e)}{N_{t_n}(e)}, for\ e \in E \quad \text{(Eq. 4)}$$

e.g. the current conversion rate, and click through rate for each competing experience are described by this quantity. It should be noted that, if $N_{t_n}(e) = 0$, then set $p_{t_n}(e)$ should be set to zero.

[0031] There are many strategies to decide which experience to show at each visit time. Depending on purposes, some of them may only use randomness. For example, the standard A/B test assigns an equal probability for each experience to show, until enough experience-reward samples are collected for conducting statistical analysis and selecting a best version. Multi armed bandit (MAB) algorithms, on the other hand, continue adjusting the strategy by balancing randomness (exploration) and current optimal choice (exploitation) based on the most recent performance, in order to achieve a higher overall average reward (including in-test rewards).

[0032] For A/B testing, the parameters needed before starting the sampling process include confidence level or type I error α, type II error β (or equivalently the power of the test $1-\beta$), and effect size (unstandardized) d, also often referred to as minimum detectable effect (MDE). Then the minimal sample sizes needed to guarantee the above specifications may be computed. Using a known formulation for running z-test (or t-test when sample size larger than 30) of comparing two sample means, where the null hypothesis is H_0: $d = 0$ and alternative hypothesis is H_1: $d \neq 0$ we can have the minimal sample sizes for the two groups, i.e., n_1 and n_2 shown in equations 5 and 6 below:

$$n_1 = \lambda n_2 \quad \text{(Eq. 5)}$$

$$n_2 = \frac{(z_{\alpha/2} + z_\beta)^2}{d^2} [p_1(1 - p_1)/\lambda + p_2(1 - p_2)] \quad \text{(Eq. 6)}$$

where $z_{\alpha/2}$ is the $(1-\alpha/2)$-th lower quantile of a standard normal distribution, and p_1 and p_2 are the true means of the two groups.

[0033] Since, in an A/B test, two groups have the same sample size, it can be assumed that $\lambda = 1$ and obtain the minimal total sample size N according to equation 7 below:

$$N = \frac{2(z_{\alpha/2}+z_{\beta})^2}{d^2}[p_1(1-p_1)+p_2(1-p_2)] \quad \text{(Eq. 7)}$$

[0034] *MAB and A/B Test Theoretical Comparison.* As noted above, A/B testing focuses on pair-wise comparisons using an equal (uniform) traffic split, but generally ignores a potentially high opportunity cost during the test (e.g., in-test rewards), while traditional MAB approaches focus on minimizing the opportunity cost (or identifying the best arm as quickly as possible), but oftentimes ends up with very unbalanced or arbitrary sample sizes over different competing experiences, thus makes the post pair-wise comparisons difficult to generate meaningful insights.

[0035] The instant application discloses two novel approaches to leverage the strengths of both MAB and A/B testing. The first one, referred to herein as "information-greedy MAB," shown in detail in Algorithm 1 below, maximizes the power of the test by maintaining the user traffic split at the optimal split point. When the ground truth success rate for different experiences falls within the conditions set forth in equation 8, the MAB algorithm will also achieve higher or equal cumulative rewards than A/B test beside the optimal test power.

$$\frac{N_{p_1}(1-p_1)}{p_1(1-p_1)+p_2(1-p_2)} \leq n_1 \leq \frac{N}{2} \quad \text{(Eq. 8)}$$

Parameters: $T \in (0, +\infty]$
Initialization: $p_{t_0}(e) = 0, \forall e \in E = \{1, 2\}$;
$H_0 = \{\text{an empty logging}\}$; $n = 0$
while $t_{n+1} < T$ **do**
  1.
  if $\exists\, e \in E = \{1, 2\}$ *s.t.* $N_{t_n}(e) = 0$ *or*
  $p_{t_n}(e)\,(1 - p_{t_n}(e)) = 0$ **then**
    $e_{t_{n+1}}$ = a random selected $e \in \{1, 2\}$
  else
    if $\frac{N_{t_n}(1)}{N_{t_n}(2)} < \sqrt{\frac{p_{t_n}(1)(1-p_{t_n}(1))}{p_{t_n}(2)(1-p_{t_n}(2))}}$ **then** $e_{t_{n+1}} = 1$;
    else if $\frac{N_{t_n}(1)}{N_{t_n}(2)} > \sqrt{\frac{p_{t_n}(1)(1-p_{t_n}(1))}{p_{t_n}(2)(1-p_{t_n}(2))}}$ **then** $e_{t_{n+1}} = 2$;
    else $e_{t_{n+1}}$ = a random selected $e \in \{1, 2\}$;
  end
  2. Observe $r_{t_{n+1}}$; H_{n+1} = concatenate $(H_n, (e_{t_{n+1}}, r_{t_{n+1}}))$
  3. $n \rightarrow n + 1$
end

Algorithm 1

[0036] As shown in algorithm 1 above, an info-greedy MAB approach may include calculating a first ratio of the total number of times each of the two versions has been provided to users, calculating a square root of a second ratio of cumulative rewards of the two versions, where the cumulative rewards for each version are calculated as a product of the cumulative reward of the version and one-minus the cumulative reward of the version, and comparing the first ratio to the square root of the second ratio.

[0037] The second approach, referred to herein as "info-reward-greedy MAB", is given in Algorithm 2 below. This second approach seeks to maximize the cumulative rewards under the constraint that its power is no less than A/B test power, when the ground truth success rate for different experiences falls within the conditions set forth in equation 8 above.

Parameters: $T \in (0, +\infty]$
Initialization: $p_{t_n}(e) = 0, \forall e \in E = \{1, 2\}$;
$H_0 = \{\text{an empty logging}\}$; $n = 0$
while $t_{n+1} < T$ **do**
  1.
  if $\exists\, e \in E = \{1, 2\}$ *s.t.* $N_{t_n}(e) = 0$ *or*
  $p_{t_n}(e)\left(1 - p_{t_n}(e)\right) = 0$ **then**
    $e_{t_{n+1}}$ = a random selected $e \in \{1, 2\}$
  else
    set $\lambda = \frac{N_{t_n}(1)}{N_{t_n}(2)}$ and $\eta = \frac{p_{t_n}(1)(1-p_{t_n}(1))}{p_{t_n}(2)(1-p_{t_n}(2))}$
    if $p_{t_n}(1) \leq p_{t_n}(2)$ **then**
      if $\lambda < \eta$ **then** $e_{t_{n+1}} = 1$;
      else if $\lambda > \eta$ **then** $e_{t_{n+1}} = 2$;
      else $e_{t_{n+1}}$ = a random selected $e \in \{1, 2\}$;
    else
      if $\lambda < \eta$ **then** $e_{t_{n+1}} = 2$;
      else if $\lambda > \eta$ **then** $e_{t_{n+1}} = 1$;
      else $e_{t_{n+1}}$ = a random selected $e \in \{1, 2\}$;
    end
  end
  2. Observe $r_{t_{n+1}}$; H_{n+1} = concatenate $\left(H_n, (e_{t_{n+1}}, r_{t_{n+1}})\right)$
  3. $n \rightarrow n + 1$
end

Algorithm 2

[0038] As shown in algorithm 2 above, an info-reward-greedy MAB approach may include calculating a first ratio of the total number of times each version has been provided to users, calculating a second ratio of cumulative rewards for the two versions, where the cumulative rewards for each version are calculated as a product of the cumulative reward of the experience and one-minus that reward, and comparing the first ratio to the second ratio.

[0039] In some embodiments, either an info-greedy MAB or an info-reward-greedy MAB may be implemented to select respective interface versions for users during the test period at block 206.

[0040] The method 200 may further include, at block 208, for each of the plurality of first users, causing the determined version of the electronic user interface to be delivered to the first user. For example, where the determined version is a particular page layout, block 208 may include causing the particular page layout to be displayed for the user (e.g., by transmitting, or causing to be transmitted, the page and the particular layout to the user computing device from which the request was received). Where the determined version is a particular set of search engine parameters, in another example, block 208 may include causing search results obtained according to those particular parameters to be delivered to the user (e.g., displayed in the interface for the user).

[0041] Blocks 204, 206, and 208 may be performed for the duration of the test period. Where the test period is a defined time period or number of users or similar, the cumulative rewards of each version may be tracked throughout the test period to enable the comparisons at block 206. Where the test period terminates when one version demonstrates a predetermined degree of superiority over other tested versions, the cumulative rewards of each version may be tracked throughout the test period, and the cumulative or average per-user rewards of the different experiences may be compared to each other on a periodic basis (e.g., after every first user's reward). In some embodiments, when the cumulative or average rewards of a given version is greater than each other version by a predetermined threshold, the test period may be terminated.

[0042] The method 200 may further include, at block 210, determining one of the two or more versions that delivered highest rewards during the test period. In some embodiments, block 210 may include determining a respective reward quantity as to each user during the test period. As noted above, a reward may be a binary or Boolean value, or may be a value form a continuous range. A reward may be indicative of whether or not—or the degree to which—the user performed a predetermined desired action. The action may be, for example, a user click on or other selection of a particular portion of the interface, a user navigation to a particular portion of the interface, a user completing a transaction through the interface, a value of a transaction completed by the user through the interface, etc. In some embodiments, block 210 may include assigning a reward value to a particular user action. For example, for a Boolean action, assigning a value may include assigning a first value to the action being performed, and a second value to the action not being performed. For a

reward value from a continuous range, assigning a value may include selecting a value from within the range based on the desirability of the user action, and/or scaling a value associated with the user action to a common scale for all rewarded actions (e.g., scaling all values to a continuous range between zero and one). Block 210 may include selecting a version that delivered highest cumulative rewards during the test period as the determined version, in some embodiments. Block 210 may include selecting a version that delivered highest average rewards during the test period as the determined version, in some embodiments.

[0043] The method 200 may further include, at block 212, receiving from a second user, after the test period, a request for the electronic user interface. The second user may be different from all of the first users, or may have been one of the first users.

[0044] The method 200 may further include, at block 214, causing the determined version that delivered highest rewards during the test period to be delivered to the second user. Delivery of the determined version may be performed in a manner similar to delivery of versions during the test period, as described above.

[0045] The approach described in the method 200 may improve upon known approaches, as described below.

[0046] *Test Results – Simulation Setup.* Extensive testing was performed to compare info-greedy MAB and info-reward-greedy MAB to various known test approaches. First, fixed-horizon testing was performed. In a fixed-horizon test comparison, all the tests end when their samples reach the same pre-specified sample size NAB or $NMAB$, which is decided by the typical A/B test requirements based on type I error α, type II error β, minimal detectable effect d, and equation 7 above. The performance between tests using the MAB algorithms and A/B testing can then be compared in terms of the power of the test results, their accuracy of identifying the best version with statistical and practical significance, and overall rewards at the end of the test period (e.g., cumulative rewards obtained during the test period). A simulation dataset was generated with uniformly random versions following a variety set of distributions, in order to test how the algorithms perform under different distribution differences. An industrial dataset was randomly selected from historical A/B tests where the traffic was uniformly randomly distributed.

[0047] *Test Results - Simulation Performance.* 6000 trials were performed under the following experimental settings. The type I error was set to 5%, MDE was 0.01, the ground truth mean of Arm 1 (i.e., version 1) is 5%, and the ground truth mean of Arm 2 (i.e., version 2) ranges from 1% to 10%. For each case, 100 rounds of offline evaluations were conducted.

As demonstrated in FIGS. 8 and 9, when the two arms have different distributions, Thompson Sampling (TS) achieves the highest total reward whereas the lowest power especially when the difference is large, due to the more aggressive traffic split. Info-greedy (I-G) and Info-reward-greedy (IR-G), on the other hand, achieve relatively high powers and larger rewards compared with the A/B test. Info-greedy (I-G) and Info-reward-greedy (IR-G) also outperformed ϵ-Greedy (ϵ-G) and UCB1 in terms of power without much loss in reward. FIGS. 3 and 4 illustrate the accuracy of each algorithms to detect a fixed winner, i.e. the percentage of trials that achieved statistical or practical significance. Info-greedy and Info-reward-greedy achieve higher accuracy in identifying the practical winners in general.

[0048] *Performance on Industrial Data Set.* Approximately 4000 trials were performed using about 40 different industrial data sets (in which the experiences delivered to the users, and the users' responsive actions, are known), the average power and the average normalized rewards for each algorithm are shown in FIG. 9. Due to the distributions of the variety of the datasets, FIG. 9 illustrates groupings based on the "true" average reward difference. Info-greedy MAB and info-reward-greedy MAB both achieve higher normalized rewards and power in all cases. UCB1 performs relatively better than A/B testing. ϵ-greedy shows the lowest power and rewards. Thompson sampling achieve similar rewards as A/B testing in the first two scenarios but higher rewards in the 3rd scenario as the true difference becomes large. However, its power is always lower than A/B testing, UCB1 and the two proposed algorithms. The probabilities to identify statistical and practical significance are almost the same and close to 0.

[0049] *Dynamic-horizon Test Comparison.* Based on the testing described above, some MAB algorithms can achieve a higher power than or equal power to A/B test given a fixed sample size and other test parameters. In reverse, this implies that to achieve a fixed test power, these MAB algorithms can use less or equal sample size relative to an A/B test. For further testing, the power was set at $1 - \beta$, and the test lengths are flexible, which end at the time they achieve the same test power under given parameters α and d. The performance between the tests using MAB algorithms and A/B testing can then be compared in terms of the total number of samples used (i.e., the speed to achieve the same test power), and their accuracy of identifying the true winner with significance. Before describing the test results, an analysis of how to define power for the tests using MAB algorithms is provided below, to ensure fairness for the comparisons with A/B test.

[0050] *Early Stopping Criterion.* A difficulty for designing flexible-length tests using MAB algorithms is that even if type I error α, type II error β (or power $1-\beta$), and minimum detectable effect d are defined, it cannot be decided in advance how many samples will be needed to achieve the requirements for a typical A/B test. This is because the final sample ratio between two groups, i.e., $\lambda = n_1, n_2$ controlled by MAB algorithms normally depend on the algorithm interactions with users' actions (e.g., rewards of those actions), where λ is usually unknown beforehand, unlike the situation of A/B testing where λ is very close to 1.

[0051] As noted above in equations 5 and 6, without knowing λ, the total sample size N needed to achieve the power and the other requirements cannot be calculated. Also without knowing N the test stopping time cannot be determined. To overcome this difficulty, the "power" of the MAB tests may be adaptively updated given the other parameter requirements (α and d). It should be noted that this approach is different from a so-called "posthoc power analysis." In the post-hoc power analysis, the unstandardized effect size d will be replaced by the sample mean difference as the test goes, however in the instant approach, throughout the process the unstandardized effect size is unchanged (i.e., a fixed MDE) and the number of samples for each experience, i.e., Ntn (1) and Ntn (2), is updated, and the variance is given by the sample variances of each experience set forth in equation 9 below:

$$S^2_{t_n}(e) = \frac{\sum_{t=1}^{n}\left[r_{t_i} - p_{t_n}(e)\right]^2 \mathbb{1}(e_{t_i}=e)}{N_{t_n}(e)-1}, e \in \{1,2\} \qquad \text{(Eq. 9)}$$

for which an unbiased estimator of the true variance for group e can be proved. A variety of numerical experiments a provided below to test this design. For the comparison fairness MAB algorithms and A/B test, all competing tests use the same updating rules for checking whether the "original" power meets the requirements.

[0052] Determining an early stopping point can be performed according to algorithm 3 below:

Algorithm 3: Aggressive Early Stopping ($|E| = 2$)

Parameters: α, β, d, S (as described in algorithm 5), $T \in (0, \infty)$

Initialization: $H_0 = \{\text{an empty logging}\}$; $\rho = 0$; $n = 0$

while $(\rho < 1-\beta)$ *and* $(t_{n+1} < T)$ **do**

1. $e_{t_{n+1}}$ = an experience generated by strategy S given H_n
2. Observe $r_{t_{n+1}}$; H_{n+1} = concatenate $(H_n, (e_{t_{n+1}}, r_{t_{n+1}}))$
3. $\rho \rightarrow \Phi\left(\frac{|d|}{\sqrt{\frac{S^2_{t_n}(1)}{\max(1, N_{t_n}(1))} + \frac{S^2_{t_n}(2)}{\max(1, N_{t_n}(2))}}} - z_{\alpha/2}\right)$
4. $n \rightarrow n+1$

end

[0053] Info-greedy MAB and info-reward-greedy MAB approaches were compared to several known test types, in addition to A/B testing, as described above. These additional tests are set forth in algorithms 4, 5, 6, and 7 below.

Algorithm 4: ϵ-greedy [15]

Parameters: $\epsilon > 0, T \in (0, +\infty]$

Initialization: $p_{t_0}(e) = 0, \forall e \in E$;
$H_0 = \{\text{an empty logging}\}; n = 0$

while $t_{n+1} < T$ **do**

1. $\sigma \longrightarrow$ Generate a uniform random number $\in [0, 1]$

if $\sigma < \epsilon$ **then** $e_{t_{n+1}}$ = a random selected $e \in E$;

$e_{t_{n+1}} = \text{argmax}_{e \in E} p_{t_n}(e)$, with random tie breaking

2. Observe $r_{t_{n+1}}$; H_{n+1} = concatenate $(H_n, (e_{t_{n+1}}, r_{t_{n+1}}))$

3. $n \longrightarrow n + 1$

end

Algorithm 5: Thompson Sampling [12]

Parameters: (dynamic) $\alpha_e, \beta_e, \forall e \in E$

Initialization: $\alpha_e > 1, \beta_e > 1, \forall e \in E$ (to avoid trivial cases); $H_0 = \{\text{an empty logging}\}; n = 0$

while $t_{n+1} < T$ **do**

1. $\forall e \in E$, $\hat{p_e} \longrightarrow$ Generate a sample from Beta distribution $B(\alpha_e, \beta_e > 1)$

2. $e_{t_{n+1}} = \underset{e \in E}{\text{argmax}}\, \hat{p_e}$, with random tie breaking

2. Observe $r_{t_{n+1}}$; H_{n+1} = concatenate $(H_n, (e_{t_{n+1}}, r_{t_{n+1}}))$

3. $(\alpha_{e_{t_{n+1}}}, \beta_{e_{t_{n+1}}}) \longrightarrow (\alpha_{e_{t_{n+1}}} + r_{t_{n+1}}, \beta_{e_{t_{n+1}}} - r_{t_{n+1}} + 1)$

4. $n \longrightarrow n + 1$

end

Algorithm 6: Upper Confidence Bound 1 (UCB1) [2]

Initialization: $p_{t_0}(e) = 0, \forall e \in E$;
$H_0 = \{\text{an empty logging}\}$; $n = 0$

while $t_{n+1} < T$ **do**

1. $e_{t_{n+1}} = \underset{e \in E}{\operatorname{argmax}}\, p_{t_n}(e) + \sqrt{\frac{2\ln(n + |E|)}{N_{t_n}(e) + 1}}$, (Adding |E| and 1 here to avoid trivial cases, ln (0) and 0 division.)
2. Observe $r_{t_{n+1}}$; $H_{n+1} = \text{concatenate}\left(H_n, (e_{t_{n+1}}, r_{t_{n+1}})\right)$
3. $n \rightarrow n + 1$

end

Algorithm 7: A/B Test Sampling

Parameters: α (type I error), β (type II error), d (MDE, substantive or practical significance),
N (total samples needed, decided by α, β, d),
$T \in (0, \infty)$

Initialization: $H_0 = \{\text{an empty logging}\}$; $n = 0$

while $(n + 1 < N)$ *and* $(t_{n+1} < T)$ **do**

1. $e_{t_{n+1}}$ = a uniformly random selected $e \in E$
2. Observe $r_{t_{n+1}}$; $H_{n+1} = \text{concatenate}\left(H_n, (e_{t_{n+1}}, r_{t_{n+1}})\right)$
3. $n \rightarrow n + 1$

end

[0054] *Simulation Performance.* With the same parameter settings as in the fixed-horizon simulations above, FIG. 5 shows the sample size used to achieve the required power as the ground truth success rate of V2 changes. Given sufficient population size, all the algorithms reached the desired power except Thompson Sampling. Info-greedy MAB and info-reward greedy MAB require similar or slightly fewer samples compared with A/B test and the sample size grows linearly as the success rate increases. A zoomed-in view presented in FIG. 6 further demonstrates that info-greedy MAB saves more samples when the distribution difference of V1 and V2 is large. UCB1 requires a similar sample size as A/B testing when the success rate is relatively small, however, it grows exponentially when the rate is larger due to more aggressive traffic split. Epsilon Greedy requires more samples but shows advantage over UCB1 when the V2 success rate is large (i.e. 0.01). The accuracy of each

algorithms to detect practical wins is similar except Thompson Sampling, which did not reach the desired test power.

[0055] *Industrial Data Performance.* For industrial data tests, the results—normalized sample sizes used to achieve the required power (0.9 here)—are shown in FIG. 5. Similarly, the test scenarios are grouped into three categories based on their "true" average reward difference between competing experiences: no less than 0 basis point (BPS), 10 BPS, and 20 BPS, respectively. The total sample size used by A/B testing was normalized to 1 as a general benchmark. As is clear from FIG. 5, both ϵ-greedy and Thompson Sampling algorithms require more samples to achieve the required power than A/B testing and this disclosure's novel approaches. UCB1 is relatively close to A/B testing. The proposed Info-greedy and Info-reward greedy algorithms use fewer samples than A/B testing. In addition, the probabilities to identify statistical and practical significance are almost the same and close to 0. As shown in FIG. 7, the info-greedy MAB approach and the info-reward-greedy MAB approach can both achieve the same test power faster than the other algorithms (including A/B test, Epsilon-greedy, Thompson Sampling, and UCB1), making it possible to stop test earlier, so information greedy MAB algorithms can shorten the testing period without power loss.

[0056] FIG. 10 is a diagrammatic view of an example embodiment of a user computing environment that includes a computing system environment 1000, such as a desktop computer, laptop, smartphone, tablet, or any other such device having the ability to execute instructions, such as those stored within a non-transient, computer-readable medium. Furthermore, while described and illustrated in the context of a single computing system, those skilled in the art will also appreciate that the various tasks described hereinafter may be practiced in a distributed environment having multiple computing systems linked via a local or wide-area network in which the executable instructions may be associated with and/or executed by one or more of multiple computing systems.

[0057] In its most basic configuration, computing system environment 1000 typically includes at least one processing unit 1002 and at least one memory 1004, which may be linked via a bus. Depending on the exact configuration and type of computing system environment, memory 1004 may be volatile (such as RAM 1010), non-volatile (such as ROM 1008, flash memory, etc.) or some combination of the two. Computing system environment 1000 may have additional features and/or functionality. For example, computing system environment 1000 may also include additional storage (removable and/or non-removable) including, but not limited to, magnetic or optical disks, tape drives and/or flash drives. Such

additional memory devices may be made accessible to the computing system environment 1000 by means of, for example, a hard disk drive interface 1012, a magnetic disk drive interface 1014, and/or an optical disk drive interface 1016. As will be understood, these devices, which would be linked to the system bus, respectively, allow for reading from and writing to a hard disk 1018, reading from or writing to a removable magnetic disk 1020, and/or for reading from or writing to a removable optical disk 1022, such as a CD/DVD ROM or other optical media. The drive interfaces and their associated computer-readable media allow for the nonvolatile storage of computer readable instructions, data structures, program modules and other data for the computing system environment 1000. Those skilled in the art will further appreciate that other types of computer readable media that can store data may be used for this same purpose. Examples of such media devices include, but are not limited to, magnetic cassettes, flash memory cards, digital videodisks, Bernoulli cartridges, random access memories, nano-drives, memory sticks, other read/write and/or read-only memories and/or any other method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Any such computer storage media may be part of computing system environment 1000.

[0058] A number of program modules may be stored in one or more of the memory/media devices. For example, a basic input/output system (BIOS) 1024, containing the basic routines that help to transfer information between elements within the computing system environment 1000, such as during start-up, may be stored in ROM 1008. Similarly, RAM 1010, hard disk 1018, and/or peripheral memory devices may be used to store computer executable instructions comprising an operating system 1026, one or more applications programs 1028 (which may include the functionality of the experience testing system 102 of FIG. 1, for example), other program modules 1030, and/or program data 1032. Still further, computer-executable instructions may be downloaded to the computing environment 1000 as needed, for example, via a network connection.

[0059] An end-user may enter commands and information into the computing system environment 1000 through input devices such as a keyboard 1034 and/or a pointing device 1036. While not illustrated, other input devices may include a microphone, a joystick, a game pad, a scanner, etc. These and other input devices would typically be connected to the processing unit 1002 by means of a peripheral interface 1038 which, in turn, would be coupled to bus. Input devices may be directly or indirectly connected to processor 1002 via interfaces such as, for example, a parallel port, game port, firewire, or a universal serial bus (USB). To view information from the computing system environment 1000, a monitor 1040

or other type of display device may also be connected to bus via an interface, such as via video adapter 1043. In addition to the monitor 1040, the computing system environment 1000 may also include other peripheral output devices, not shown, such as speakers and printers.

[0060] The computing system environment 1000 may also utilize logical connections to one or more computing system environments. Communications between the computing system environment 1000 and the remote computing system environment may be exchanged via a further processing device, such a network router 1042, that is responsible for network routing. Communications with the network router 1042 may be performed via a network interface component 1044. Thus, within such a networked environment, e.g., the Internet, World Wide Web, LAN, or other like type of wired or wireless network, it will be appreciated that program modules depicted relative to the computing system environment 1000, or portions thereof, may be stored in the memory storage device(s) of the computing system environment 1000.

[0061] The computing system environment 1000 may also include localization hardware 1046 for determining a location of the computing system environment 1000. In embodiments, the localization hardware 1046 may include, for example only, a GPS antenna, an RFID chip or reader, a WiFi antenna, or other computing hardware that may be used to capture or transmit signals that may be used to determine the location of the computing system environment 1000.

[0062] The computing environment 1000, or portions thereof, may comprise one or more components of the system 100 of FIG. 1, in embodiments.

[0063] While this disclosure has described certain embodiments, it will be understood that the claims are not intended to be limited to these embodiments except as explicitly recited in the claims. On the contrary, the instant disclosure is intended to cover alternatives, modifications and equivalents, which may be included within the spirit and scope of the disclosure. Furthermore, in the detailed description of the present disclosure, numerous specific details are set forth in order to provide a thorough understanding of the disclosed embodiments. However, it will be obvious to one of ordinary skill in the art that systems and methods consistent with this disclosure may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure various aspects of the present disclosure.

[0064] Some portions of the detailed descriptions of this disclosure have been presented in terms of procedures, logic blocks, processing, and other symbolic representations of

operations on data bits within a computer or digital system memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. A procedure, logic block, process, etc., is herein, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these physical manipulations take the form of electrical or magnetic data capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system or similar electronic computing device. For reasons of convenience, and with reference to common usage, such data is referred to as bits, values, elements, symbols, characters, terms, numbers, or the like, with reference to various embodiments of the present invention.

[0065] It should be borne in mind, however, that these terms are to be interpreted as referencing physical manipulations and quantities and are merely convenient labels that should be interpreted further in view of terms commonly used in the art. Unless specifically stated otherwise, as apparent from the discussion herein, it is understood that throughout discussions of the present embodiment, discussions utilizing terms such as “determining” or “outputting” or “transmitting” or “recording” or “locating” or “storing” or “displaying” or “receiving” or “recognizing” or “utilizing” or “generating” or “providing” or “accessing” or “checking” or “notifying” or “delivering” or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data. The data is represented as physical (electronic) quantities within the computer system’s registers and memories and is transformed into other data similarly represented as physical quantities within the computer system memories or registers, or other such information storage, transmission, or display devices as described herein or otherwise understood to one of ordinary skill in the art.

Claims

What is claimed is:

1. A method for determining a user experience for an electronic user interface, the method comprising:

   defining a test period for testing two or more versions of an electronic user interface;

   receiving, from each of a plurality of users during the test period, a respective request for the electronic user interface;

   determining, for each of the plurality of users, a respective version of the two or more versions of the electronic user interface by maximizing test power during the test period while maintaining higher in-test rewards than an A/B test; and

   causing, for each of the plurality of users, the determined version of the electronic user interface to be delivered to the user.

2. The method of claim 1, wherein the two or more versions comprises two versions, wherein maximizing test power during the test period while maintaining higher in-test rewards than an A/B test comprises:

   calculating a first ratio of a total number of times each of the two versions has been provided to users;

   calculating a square root of a second ratio of cumulative rewards of the two versions, where the cumulative rewards for each version are calculated as a product of the cumulative reward of the version and one-minus the cumulative reward of the version; and

   comparing the first ratio to the square root of the second ratio.

3. The method of claim 2, wherein the reward is a Boolean value.

4. The method of claim 3, wherein the reward for a user indicates whether the user performed a predefined action in the electronic user interface.

5. The method of claim 2, wherein the reward is a continuous value between zero and one.

6. The method of claim 5, wherein the reward indicates a value from a continuous range of values of an interaction of the user with the electronic user interface.

7. The method of claim 1, wherein the electronic user interface is a portion of a webpage.

8. The method of claim 7, wherein the portion of the webpage comprises a home page of a website.

9. The method of claim 1, wherein the users are first users, the method further comprising:

determining one of the two or more versions that delivered highest rewards during the test period;

after the test period, receiving a request from a second user for the electronic user interface; and

causing the determined version that delivered highest rewards during the test period to be delivered to the second user.

10. A method for determining a user experience for an electronic user interface, the method comprising:

defining a test period for testing two or more versions of an electronic user interface;

receiving, from each of a plurality of users during the test period, a respective request for the electronic user interface;

determining, for each of the plurality of users, a respective version of the two or more versions of the electronic user interface by maximizing rewards during the test period while maintaining a test power no worse than an A/B test; and

causing, for each of the plurality of users, the determined version of the electronic user interface to be delivered to the user.

11. The method of claim 10, wherein the two or more versions comprises two versions, wherein maximize the rewards during the test period while maintaining a test power no worse than an A/B test comprises:

calculating a first ratio of a total number of times each version has been provided to users;

calculating a second ratio of cumulative rewards for the two versions, where the cumulative rewards for each version are calculated as a product of the cumulative reward of the experience and one-minus that reward; and

comparing the first ratio to the second ratio.

12. The method of claim 11, wherein the reward is a Boolean value.

13. The method of claim 12, wherein the reward for a user indicates whether the user performed a predefined action in the electronic user interface.

14. The method of claim 11, wherein the reward is a continuous value between zero and one.

15. The method of claim 14, wherein the reward indicates a value from a continuous range of values of an interaction of the user with the electronic user interface.

16. The method of claim 10, wherein the electronic user interface is a portion of a webpage.

17. The method of claim 16, wherein the portion of the webpage comprises a home page of a website.

18. The method of claim 10, wherein the users are first users, the method further comprising:

determining one of the two or more versions that delivered highest rewards during the test period;

after the test period, receiving a request from a second user for the electronic user interface; and

causing the determined version that delivered highest rewards during the test period to be delivered to the second user.

19. A method for determining a user experience for an electronic user interface, the method comprising:

defining a test period for testing two or more versions of an electronic user interface;

receiving, from each of a plurality of users during the test period, a respective request for the electronic user interface;

determining, for each of the plurality of users, a respective version of the two or more versions of the electronic user interface by:

maximizing test power during the test period while maintaining higher in-test rewards than an A/B test; or

maximizing the rewards during the test period while maintaining a test power no worse than an A/B test; and

causing, for each of the plurality of users, the determined version of the electronic user interface to be delivered to the user.

20. The method of claim 19, wherein the two or more versions comprises two versions, wherein:

maximizing the rewards during the test period while maintaining a test power no worse than an A/B test comprises:

calculating a first ratio of a total number of times each version has been provided to users;

calculating a second ratio of cumulative rewards for the two versions, where the cumulative rewards for each version are calculated as a product of the cumulative reward of the experience and one-minus that reward; and

comparing the first ratio to the second ratio; and

maximizing test power during the test period while maintaining higher in-test rewards than an A/B test comprises:

calculating a first ratio of the total number of times each of the two versions has been provided to users;

calculating a square root of a second ratio of cumulative rewards of the two versions, where the cumulative rewards for each version are calculated as a product of the cumulative reward of the version and one-minus the cumulative reward of the version; and

comparing the first ratio to the square root of the second ratio.

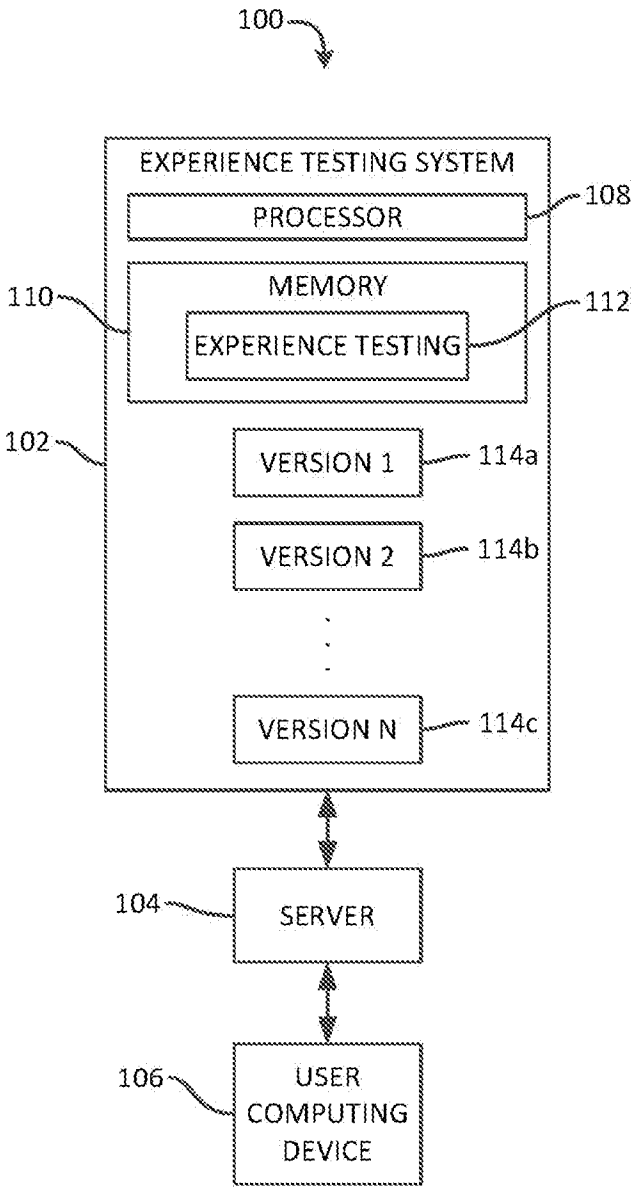
100
EXPERIENCE TESTING SYSTEM
PROCESSOR
108
MEMORY
110
112
EXPERIENCE TESTING
102
VERSION 1
114a
VERSION 2
114b
VERSION N
114c
104
SERVER
106
USER COMPUTING DEVICE

FIG. 1

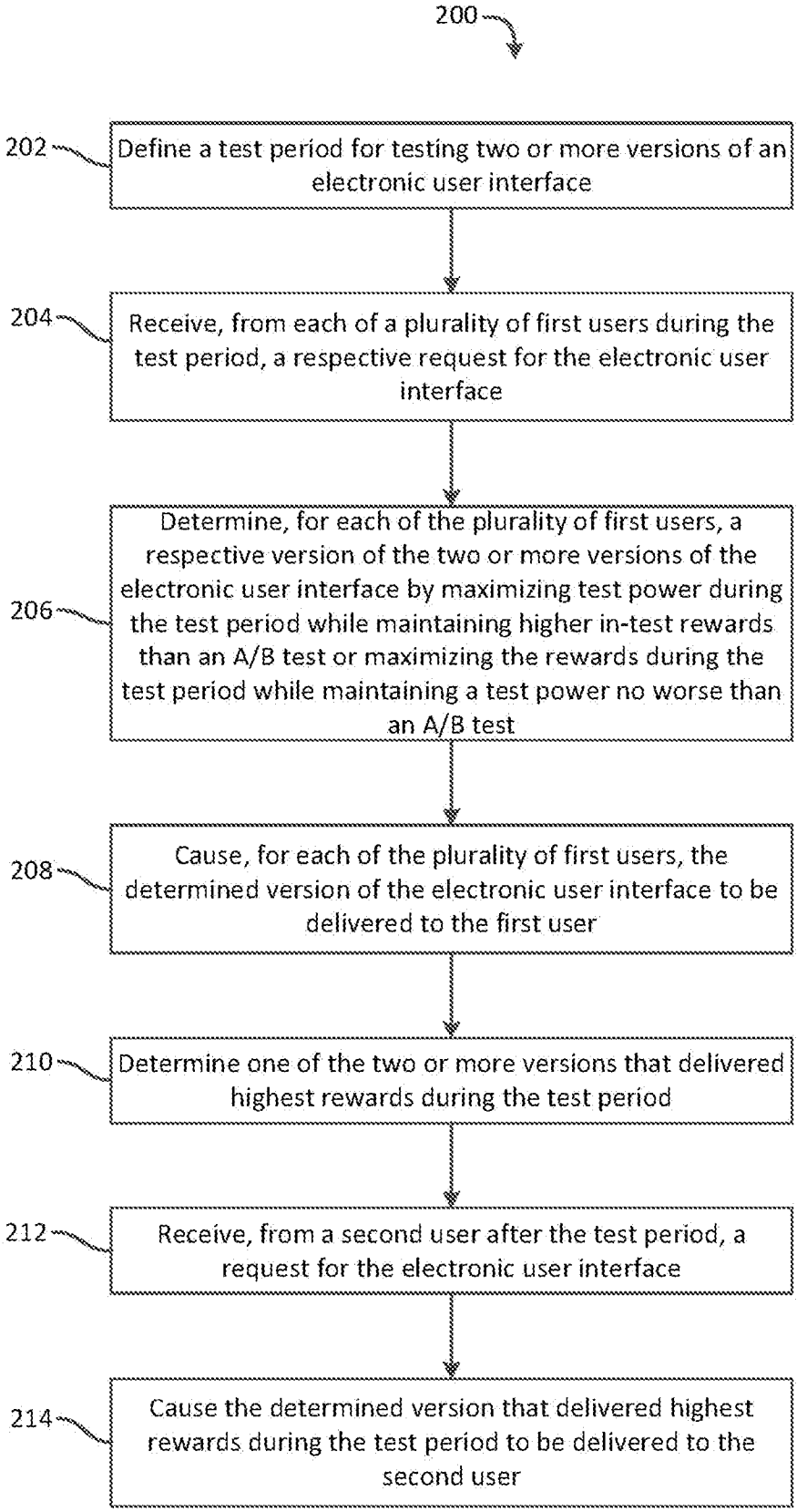
200
202
Define a test period for testing two or more versions of an electronic user interface
204
Receive, from each of a plurality of first users during the test period, a respective request for the electronic user interface
206
Determine, for each of the plurality of first users, a respective version of the two or more versions of the electronic user interface by maximizing test power during the test period while maintaining higher in-test rewards than an A/B test or maximizing the rewards during the test period while maintaining a test power no worse than an A/B test
208
Cause, for each of the plurality of first users, the determined version of the electronic user interface to be delivered to the first user
210
Determine one of the two or more versions that delivered highest rewards during the test period
212
Receive, from a second user after the test period, a request for the electronic user interface
214
Cause the determined version that delivered highest rewards during the test period to be delivered to the second user

FIG. 2

Table 1: Statistical Wins for Fixed-horizon Simulations

$v_1 = 0.05$ $v_2 =$	A/B	ϵ-G	I-G	IR-G	TS	UCB1
0.03	100%	99%	100%	100%	96%	100%
0.04	85%	56%	81%	79%	62%	80%
0.05	1%	6%	5%	5%	8%	3%
0.06	82%	54%	88%	86%	62%	90%
0.07	100%	100%	100%	100%	95%	100%
o/w	100%	100%	100%	100%	100%	100%

FIG. 3

Table 2: Practical Wins for Fixed-horizon Simulations

$v_1 = 0.05$ $v_2 =$	prac. sig.	A/B	ϵ-G	I-G	IR-G	TS	UCB1
0.01	0.03	84%	68%	93%	92%	42%	85%
0.02	0.02	92%	57%	93%	92%	32%	89%
0.03	0.01	81%	52%	90%	88%	38%	85%
0.04	0.005	22%	16%	21%	28%	20%	28%
0.05	0.005	0%	0%	0%	0%	0%	0%
0.06	0.005	23%	18%	21%	19%	24%	19%
0.07	0.01	88%	46%	89%	83%	40%	80%
0.08	0.02	85%	42%	86%	89%	15%	78%
0.09	0.03	83%	46%	88%	91%	24%	67%
0.10	0.04	92%	50%	84%	85%	17%	61%

FIG. 4

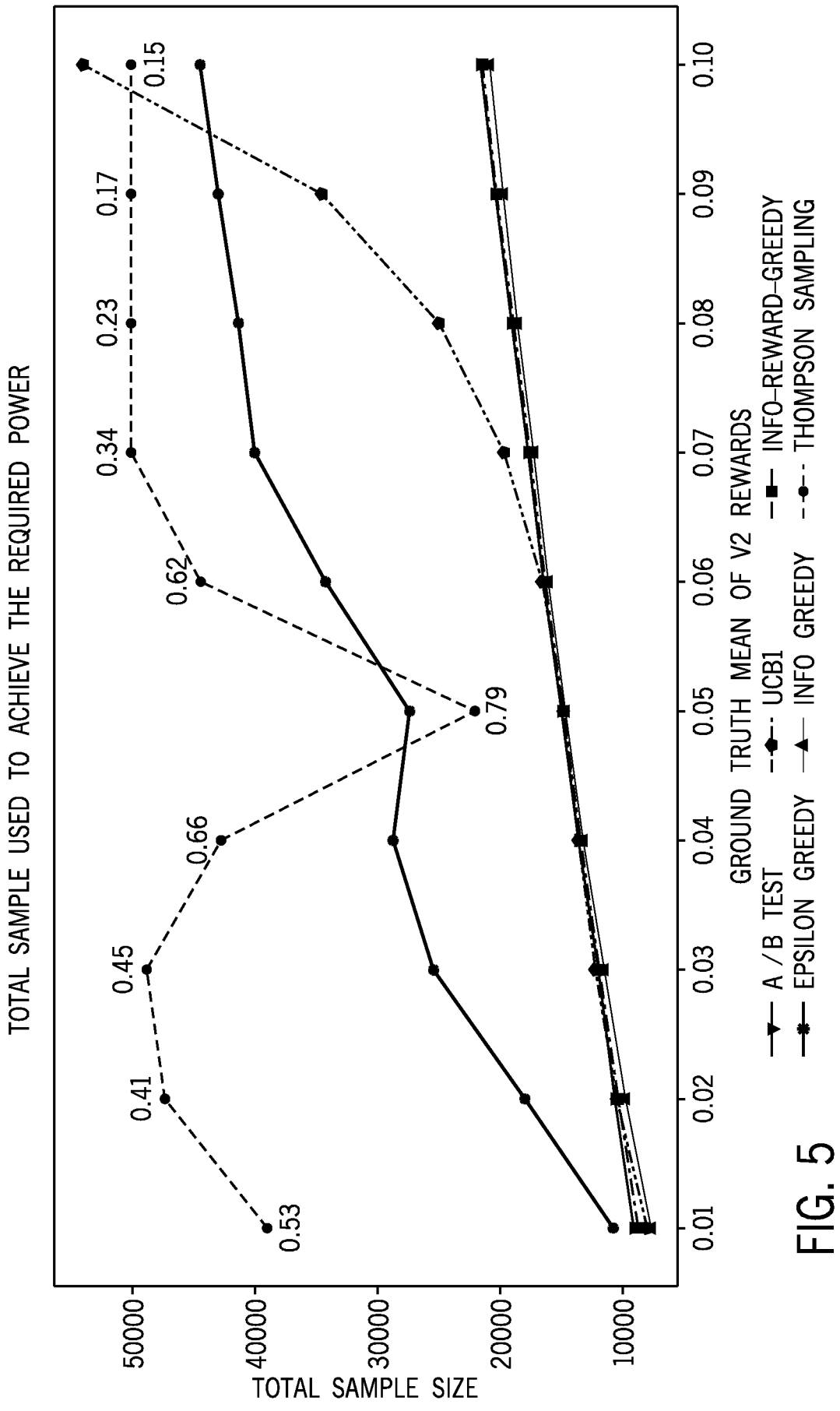
TOTAL SAMPLE USED TO ACHIEVE THE REQUIRED POWER
TOTAL SAMPLE SIZE
50000
40000
30000
20000
10000
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
GROUND TRUTH MEAN OF V2 REWARDS
A / B TEST
UCB1
INFO–REWARD–GREEDY
EPSILON GREEDY
INFO GREEDY
THOMPSON SAMPLING
0.53
0.41
0.45
0.66
0.79
0.62
0.34
0.23
0.17
0.15

FIG. 5

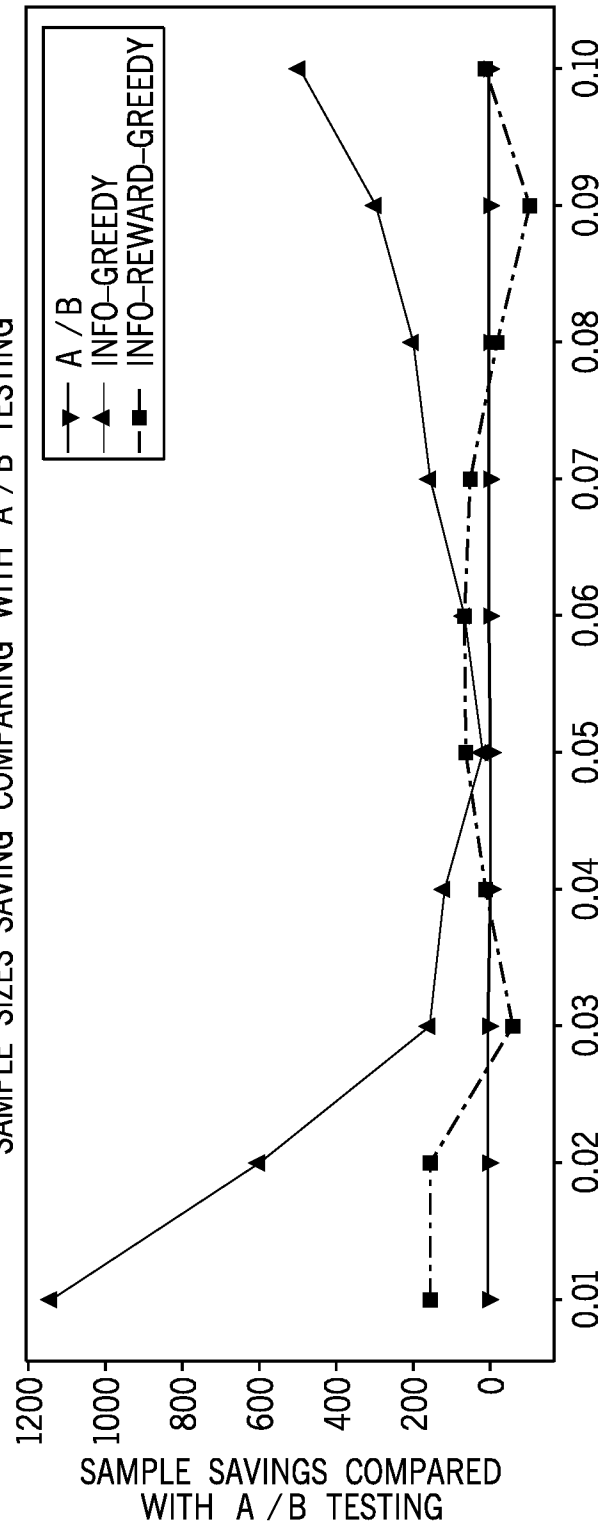
SAMPLE SIZES SAVING COMPARING WITH A/B TESTING
A/B
INFO-GREEDY
INFO-REWARD-GREEDY
SAMPLE SAVINGS COMPARED WITH A/B TESTING
1200
1000
800
600
400
200
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
GROUND TRUTH MEAN OF V2 REWARDS

FIG. 6

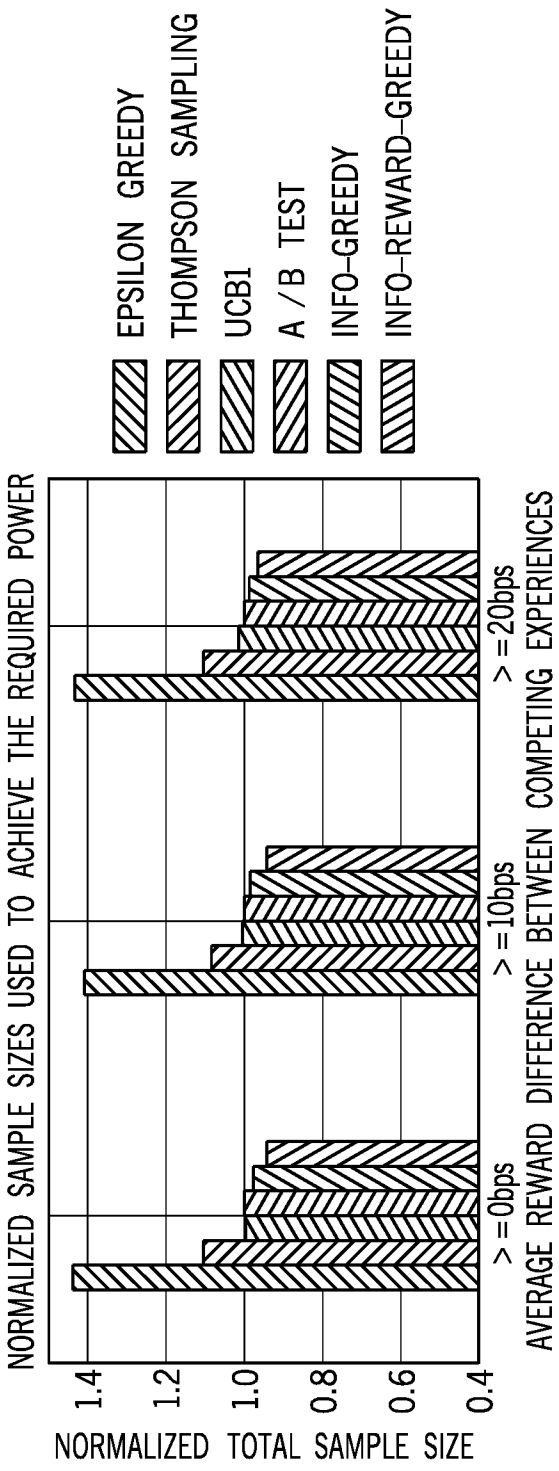
NORMALIZED SAMPLE SIZES USED TO ACHIEVE THE REQUIRED POWER
EPSILON GREEDY
THOMPSON SAMPLING
UCB1
A/B TEST
INFO-GREEDY
INFO-REWARD-GREEDY
NORMALIZED TOTAL SAMPLE SIZE
1.4
1.2
1.0
0.8
0.6
0.4
>=0bps
>=10bps
>=20bps
AVERAGE REWARD DIFFERENCE BETWEEN COMPETING EXPERIENCES

FIG. 7

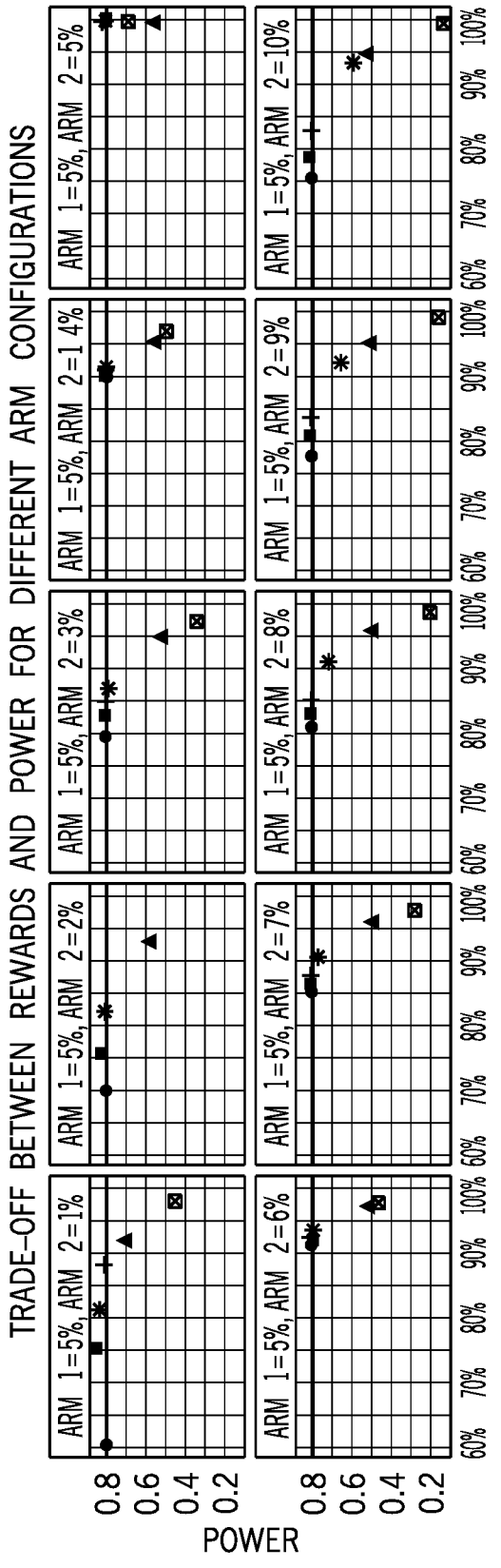
TRADE-OFF BETWEEN REWARDS AND POWER FOR DIFFERENT ARM CONFIGURATIONS
ARM 1=5%, ARM 2=1%
ARM 1=5%, ARM 2=2%
ARM 1=5%, ARM 2=3%
ARM 1=5%, ARM 2=1 4%
ARM 1=5%, ARM 2=5%
ARM 1=5%, ARM 2=6%
ARM 1=5%, ARM 2=7%
ARM 1=5%, ARM 2=8%
ARM 1=5%, ARM 2=9%
ARM 1=5%, ARM 2=10%
POWER
0.8
0.6
0.4
0.2
60%
70%
80%
90%
100%
NORMALIZED REWARDS

FIG. 8

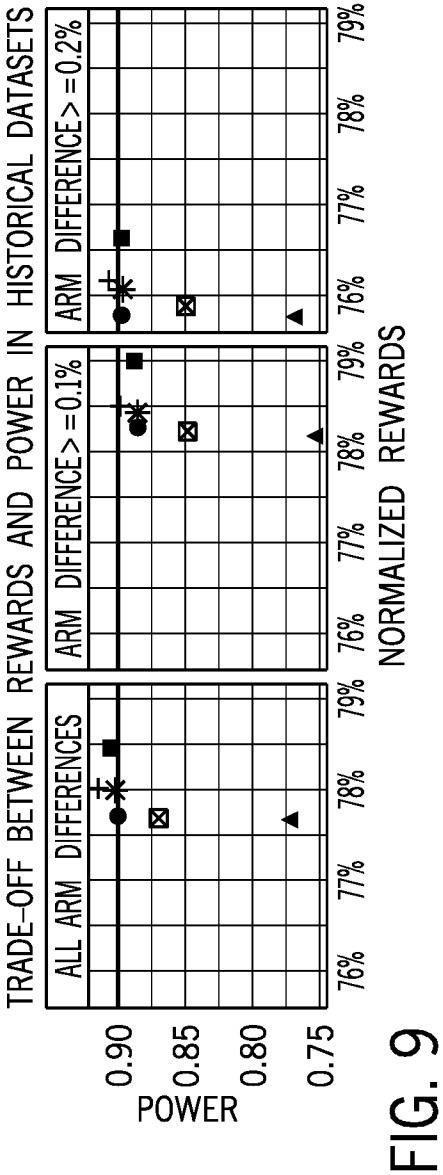
TRADE-OFF BETWEEN REWARDS AND POWER IN HISTORICAL DATASETS
ALL ARM DIFFERENCES
ARM DIFFERENCE>=0.1%
ARM DIFFERENCE>=0.2%
TEST DESIGN
● A / B TEST
▲ EPSILON GREEDY
⊠ THOMPSON SAMPLING
✱ UCB1
■ INFO-GREEDY
+ INFO-REWARD-GREEDY
POWER
0.90
0.85
0.80
0.75
76%
77%
78%
79%
NORMALIZED REWARDS

FIG. 9

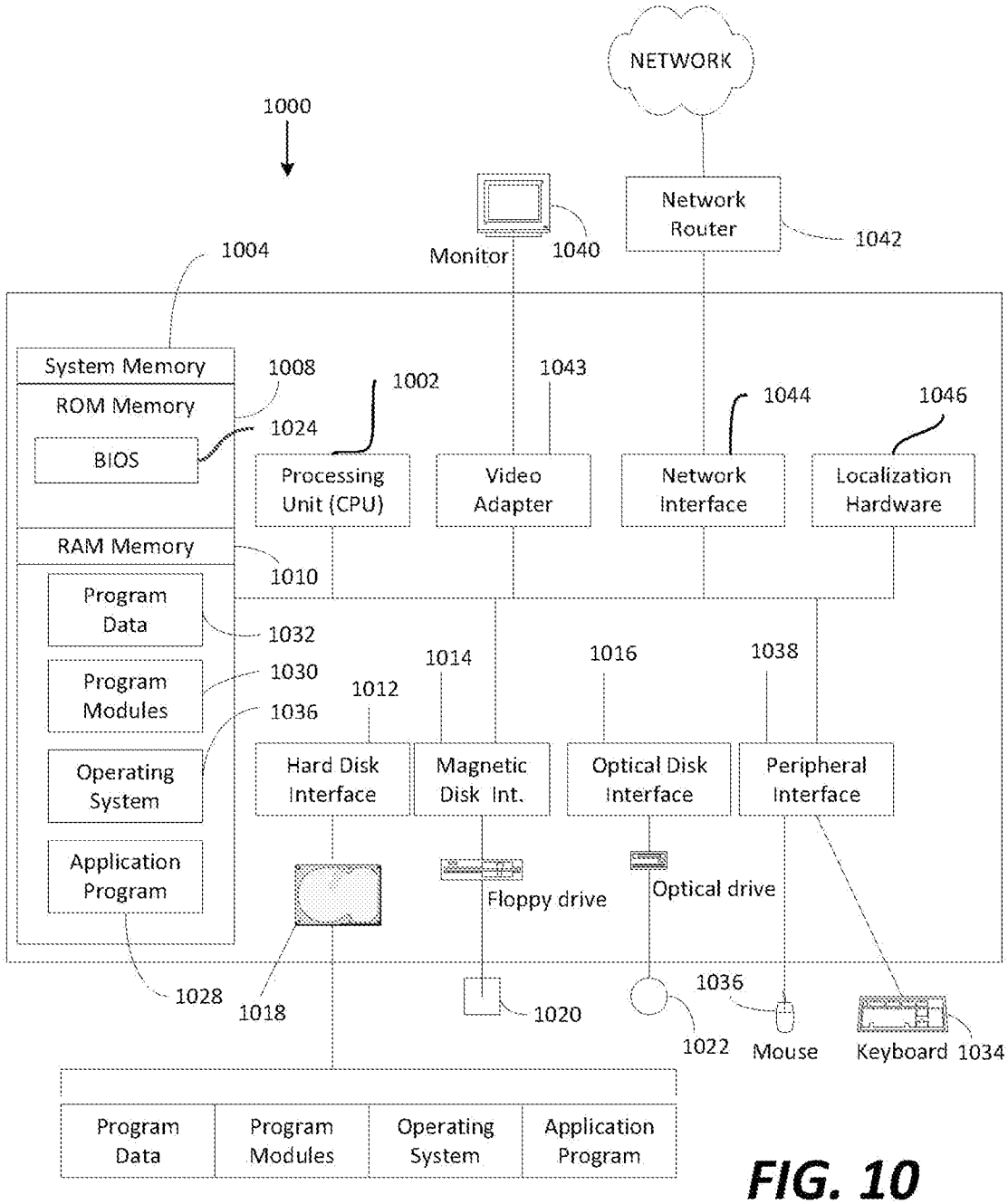
NETWORK
1000
Monitor
1040
Network Router
1042
1004
System Memory
1008
ROM Memory
1024
BIOS
1002
1043
1044
1046
Processing Unit (CPU)
Video Adapter
Network Interface
Localization Hardware
RAM Memory
1010
Program Data
1032
Program Modules
1030
Operating System
1036
Application Program
1012
1014
1016
1038
Hard Disk Interface
Magnetic Disk Int.
Optical Disk Interface
Peripheral Interface
Floppy drive
Optical drive
1028
1018
1020
1036
1022
Mouse
Keyboard
1034
Program Data
Program Modules
Operating System
Application Program

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2023/012611

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - INV. - G06Q 30/0203; G06F 11/36; G06Q 30/0242; H04L 67/53 (2023.01)
ADD. - G06F 11/34 (2023.01)

CPC - INV. - G06Q 30/0203; G06F 11/3684; G06F 16/9566; G06Q 30/0243; H04L 67/535 (2023.02)
ADD. - G06F 11/3476 (2023.02)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document

Electronic database consulted during the international search (name of database and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2020/0342500 A1 (CAPITAL ONE SERVICES LLC) 29 October 2020 (29.10.2020) entire document	1-20
A	US 2019/0244110 A1 (COGNIZANT TECHNOLOGY SOLUTIONS US CORPORATION) 08 August 2019 (08.08.2019) entire document	1-20
A	US 2015/0356103 A1 (AMERICAN EXPRESS TRAVEL RELATED SERVICES COMPANY INC) 10 December 2015 (10.12.2015) entire document	1-20
A	US 2020/0342043 A1 (OPTIMIZELY INC) 29 October 2020 (29.10.2020) entire document	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 April 2023	JUN 06 2023
Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300	Authorized officer Taina Matos Telephone No. PCT Helpdesk: 571-272-4300