Safe Online Bid Optimization with Uncertain ROI and Budget Constraints

Matteo Castiglioni
Politecnico di Milano
Milano, Italy
matteo.castiglioni@polimi.it

Alessandro Nuara
MLcube
Milano, Italy
alessandro.nuara@mlcube.com

Giulia Romano
Politecnico di Milano
Milano, Italy
giulia.romano@polimi.it

Giorgio Spadaro
Politecnico di Milano
Milano, Italy
giorgio.spadaro@mail.polimi.it

Francesco Trovò
Politecnico di Milano
Milano, Italy
francesco1.trovo@polimi.it

Nicola Gatti
Politecnico di Milano
Milano, Italy
nicola.gatti@polimi.it

ABSTRACT

In online advertising, the advertiser's goal is usually a tradeoff between achieving *high volumes* and *high profitability*. The companies' business units customarily address this tradeoff by maximizing the volumes while guaranteeing a minimum Return On Investment (ROI). This paper investigates combinatorial bandit algorithms for the bid optimization of advertising campaigns subject to *uncertain* budget and ROI constraints. We show that the problem is inapproximable within any factor unless P = NP even without uncertainty, and we provide a pseudo-polynomial-time algorithm that achieves an optimal solution. Furthermore, we show that no online learning algorithm can violate the (budget or ROI) constraints during the learning process a sublinear number of times while guaranteeing a sublinear pseudo-regret. We provide the $\mathsf{GCB}_{\mathsf{safe}}$ algorithm guaranteeing w.h.p. a constant upper bound on the number of constraints violations at the cost of a linear pseudo-regret bound. However, a simple adaptation of $\mathsf{GCB}_{\mathsf{safe}}$ provides a sublinear pseudo-regret when accepting the satisfaction of the constraints with a fixed tolerance. Finally, we experimentally evaluate $\mathsf{GCB}_{\mathsf{safe}}$ in terms of pseudo-regret/constraint-violation tradeoff in settings generated from real-world data.

KEYWORDS

Regret Minimization; Online Learning; Safe Online Learning; Uncertain Constraints; Advertising

1 INTRODUCTION

Nowadays, Internet advertising is the leading advertising medium. Notably, while the expenditure on physical ads, radio, and television has been stable for a decade, that on Internet advertising is increasing with an average ratio of 20% per year, reaching the considerable amount of 124 billion USD in 2019 only in the US [15]. Internet advertising has two main advantages over traditional advertising channels. The former is to provide a precise ad targeting, and the latter is to allow an accurate evaluation of investment performance. On the other hand, the amount of data provided by the platforms and the plethora of parameters to be set make its optimization impractical without AI tools.

Appears at the 1st Workshop on Learning with Strategic Agents (LSA 2022). Held as part of the Workshops at the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), N. Bishop, M. Han, L. Tran-Thanh, H. Xu, H. Zhang (chairs), May 9–10, 2022, Online.

The advertiser's goal is to set bids to balance the tradeoff between achieving *high volumes*, maximizing the sales of the products to advertise, and *high profitability*, maximizing ROI. The companies' business units need simple ways to address this tradeoff, and, usually, they maximize the volumes while constraining the ROI to be above a threshold. The analysis of data on the auctions on Google's AdX by Golrezaei et al. [13] shows that many advertisers have ROI constraints, particularly in hotel booking, *e.g.*, on Google Hotels. However, most of the platforms do not provide any feature to force the satisfaction of these constraints, which are *uncertain* as the revenues and costs are *a priori* unknown. Thus, the bidders need to develop bidding strategies (usually referred to as *safe*) satisfying these uncertain constraints during the entire learning process. In particular, the violation of constraints in the early stages, whose nature is almost purely explorative, can worry the bidders and be a concrete obstacle to the adoption of algorithms in this field. Our paper investigates bidding algorithms when ROI constraints and, potentially, budget constraints (*e.g.*, when the daily budget limit cannot be set on the platform) are uncertain, providing theoretical guarantees on pseudo-regret and safety.

Related Works. Many works study Internet advertising, both from the *publisher* perspective (*e.g.*, Vazirani et al. [29] design auctions for ads allocation and pricing) and from the *advertiser* perspective (*e.g.*, Feldman et al. [10] study the budget optimization problem in search advertising). Few works deal with ROI constraints, and, to the best of our knowledge, they only focus on the auction mechanisms (*e.g.*, Szymanski and Lee [26] and Borgs et al. [4] show that ROI-based bidding heuristics lead to cyclic behavior and reduce the allocation's efficiency, while Golrezaei et al. [13] propose more efficient auctions with ROI constraints). Existing learning algorithms for daily bid optimization address only budget constraints in the restricted case in which the platform allows the advertisers to set a daily budget limit (notice that some platforms such as, *e.g.*, TripAdvisor and Trivago, do not even allow the setting of the daily budget limit). For instance, Zhang et al. [30] provide an *offline* algorithm that exploits accurate models of the campaigns' performance based on low-level data, which are rarely available to the advertisers. Nuara et al. [20] provide an *online* learning algorithm that combines combinatorial multi-armed bandit techniques [6] with regression by Gaussian Processes [23]. More recent works also present pseudo-regret bounds [21], and study subcampaigns interdependencies [19]. Thomaidou et al. [27] provide a genetic algorithm for budget optimization of advertising campaigns. [9] and [28] address the bid optimization problem in a single subcampaign scenario when the budget constraint is cumulative over time.

Very recent works study bandit problems with safe exploration, in which the constraints are uncertain, and the goal is to guarantee w.h.p. their satisfaction during the entire learning process. However, the only known results are for continuous and convex arm spaces and convex constraints. In such settings, the learner can achieve the optimal solution without violating the constraints [2, 18]. Conversely, the case with discrete and/or non-convex arm spaces or non-convex constraints, such as ours, is unexplored in the literature so far. Some bandit algorithms address uncertain constraints where the goal is their satisfaction on average [5, 17]. However, the per-round violation can be arbitrarily large, and this does not fit with our setting, as the advertisers could be alarmed and, thus, give up on adopting the algorithm. Several other works in the reinforcement learning [12, 14, 22] and multi-armed bandit [11, 25] fields investigate safe exploration, providing safety guarantees on the revenue provided by the algorithm, but not on the satisfaction w.h.p. of uncertain constraints.

Original Contributions. As customary in the literature, see, *e.g.*, Devanur and Kakade [8], we make the assumption of stochastic (*i.e.*, non-adversarial) clicks, and we adopt Gaussian Processes (GPs) to model the problem parameters.[1] We show that no approximation within any strictly positive factor is possible with ROI and budget constraints unless P = NP, even in simple instances when all the parameter values are known. However, when dealing with a discretized space of the bids as it happens in practice, the problem admits an exact pseudo-polynomial time algorithm based on dynamic programming. Remarkably, we prove that, in cases beyond those with continuous and convex arm spaces and convex constraints, no online learning algorithm can violate the uncertain constraints a sublinear number of times while guaranteeing a sublinear pseudo-regret (this result holds in generic bandit settings with uncertain constraints beyond advertising). We show that a sublinear pseudo-regret can be obtained by adopting the GCB algorithm proposed by Accabi et al. [1], and we propose a novel algorithm, called $\text{GCB}_{\texttt{safe}}$, guaranteeing w.h.p. a constant upper bound on the number of constraints' violations. Most interestingly, when accepting a tolerance ψ in the satisfaction of the constraints, a simple adaptation of $\text{GCB}_{\texttt{safe}}$, namely $\text{GCB}_{\texttt{safe}}(\psi)$, guarantees both the violation w.h.p. of the constraints for a constant number of times and a sublinear pseudo-regret $O\left(\sqrt{T\sum_{j=1}^{N}\gamma_{j,T}}\right)$, where T is the time horizon of the learning process, and $\gamma_{j,T}$ is the maximum information gain of the GP used to model the j-th advertising subcampaign. Finally, we experimentally evaluate the performance of our algorithms, showing the tradeoff between pseudo-regret and constraint-violation with realistic settings generated from real-world data.

[1] The assumption that clicks are generated stochastically is reasonable in practice as advertising platforms can limit manipulation due to malicious bidders. For instance, Google Ads can identify invalid clicks and exclude them from the advertisers' spending.

2 PROBLEM FORMULATION

We are given an advertising campaign $C = \{C_1, \dots, C_N\}$, with $N \in \mathbb{N}$, where C_j is the j-th subcampaign, and a finite time horizon of $T \in \mathbb{N}$ rounds (each corresponding to one day in our application). In this work, as common in the literature on ad allocation optimization, we refer to a subcampaign as a single ad or a group of homogeneous ads requiring to set the same bid. For each day $t \in \{1, \dots, T\}$ and for every subcampaign C_j, the advertiser needs to specify the bid $x_{j,t} \in X_j$, where $X_j \subset \mathbb{R}^+$ is a finite set of bids we can set in subcampaign C_j. The goal is, for every day $t \in \{1, \dots, T\}$, to find the values of bids that maximize the overall cumulative expected revenue while keeping the overall ROI above a fixed value $\lambda \in \mathbb{R}^+$ and the overall budget below a daily value $\beta \in \mathbb{R}^+$. Formally, the resulting constrained optimization problem at day t is as follows:

$$\max_{(x_{1,t},\dots,x_{N,t})\in X_1\times\dots\times X_N} \sum_{j=1}^{N} v_j\, n_j(x_{j,t}) \tag{1a}$$

$$\text{s.t.} \quad \frac{\sum_{j=1}^{N} v_j\, n_j(x_{j,t})}{\sum_{j=1}^{N} c_j(x_{j,t})} \geq \lambda, \tag{1b}$$

$$\sum_{j=1}^{N} c_j(x_{j,t}) \leq \beta, \tag{1c}$$

where $n_j(x_{j,t})$ and $c_j(x_{j,t})$ are the expected number of clicks and the expected cost given the bid $x_{j,t}$ for subcampaign C_j, respectively, and v_j is the value per click for subcampaign C_j. Moreover, Constraint (1b) is the ROI constraint, forcing the revenue to be at least λ times the costs, and Constraint (1c) keeps the daily spend under a predefined overall budget β.[2]

In our online learning setting, $n_j(\cdot)$ and $c_j(\cdot)$ are unknown functions that we need to estimate within the time horizon T, whereas the available arms are the different values of the bid $x_{j,t} \in X_j$ satisfying the combinatorial constraints of the optimization problem.[3] A super-arm is a profile specifying one bid per subcampaign. A learning policy $\mathfrak{U}$ solving such a problem is an algorithm returning, for each day t, a set of bid $\left\{\hat{x}_{j,t}\right\}_{j=1}^{N}$. The policy $\mathfrak{U}$ can only use estimates of the unknown number-of-click and cost functions built during the learning process. Therefore, the returned solutions may not be optimal and/or violate Constraints (1b) and (1c) computed on the true functions. Notice that, even if this setting is closely related to the one presented in the work by Badanidiyuru et al. [3], the specific non-matroidal nature of the constraints do not allow to cast the bid allocation problem above into the bandit with knapsack framework.

We are interested in evaluating learning policies $\mathfrak{U}$ in terms of both loss of revenue (a.k.a. pseudo-regret) and violation of those constraints. The pseudo-regret and safety of a learning policy $\mathfrak{U}$ are defined as follows:

[2] In economic literature, it is also used an alternative definition of ROI: $\frac{\sum_{j=1}^{N}\left[v_j\, n_j(x_{j,t}) - c_j(x_{j,t})\right]}{\sum_{j=1}^{N} c_j(x_{j,t})}$. To capture this case, it is sufficient to substitute the right hand side of Constraint (1b) with $\lambda + 1$.

[3] Here, we assume that the value per click v_j is known. In the case one needs its estimates, refer to Nuara et al. [20] for details.

Algorithm 1 Meta-algorithm

Input: sets X_j of bid values, ROI threshold λ, daily budget β

Initialize the GPs for the number of clicks and costs
for $t \in \{1, \ldots, T\}$ **do**
  for $j \in \{1, \ldots, N\}$ **do**
    for $x \in X_j$ **do**
      Produce estimates $\hat{n}_{j,t-1}(x)$, $\hat{\sigma}^n_{j,t-1}(x)$ using the GP on the number of clicks
      Produce estimates $\hat{c}_{j,t-1}(x)$, $\hat{\sigma}^c_{j,t-1}(x)$ using the GP on the costs
  Compute $\boldsymbol{\mu}$ using the GPs estimates
  Run the $\text{Opt}(\boldsymbol{\mu}, \lambda)$ procedure to get a solution $\left\{\hat{x}_{j,t}\right\}_{j=1}^N$
  Set the prescribed allocation during day t
  Get revenue $\sum_{j=1}^N v_j \ \tilde{n}_j(\hat{x}_{j,t})$
  Update the GPs using the new information $\tilde{n}_{j,t}(\hat{x}_{j,t})$ and $\tilde{c}_{j,t}(\hat{x}_{j,t})$

Definition 1 (Learning policy pseudo-regret). *Given a learning policy $\mathfrak{U}$, we define the* pseudo-regret *as:*

$$R_T(\mathfrak{U}) := T\,G^* - \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{N} v_j \ n_j(\hat{x}_{j,t})\right],$$

where $G^* := \sum_{j=1}^N v_j \ n_j(x_j^*)$ is the expected revenue provided by a clairvoyant algorithm, the set of bids $\left\{x_j^*\right\}_{j=1}^N$ is the optimal clairvoyant solution to the problem in Equations (1a)–(1c)*, and the expectation $\mathbb{E}[\,\cdot\,]$ is taken w.r.t. the stochasticity of the learning policy $\mathfrak{U}$.*

Our goal is the design of algorithms that minimize the pseudo-regret $R_T(\mathfrak{U})$. In particular, we are interested in *no-regret* algorithms guaranteeing a regret that increases sublinearly in T.

Definition 2 (η-safe learning policy). *Given $\eta \in (0,\ T]$, a learning policy $\mathfrak{U}$ is η-*safe *if $\left\{\hat{x}_{j,t}\right\}_{j=1}^N$*, i.e., *the expected number of times at least one of the Constraints* (1b) *and* (1c) *is violated from $t = 1$ to T is less than η or, formally:*

$$\sum_{t=1}^{T}\mathbb{P}\left(\frac{\sum_{j=1}^N v_j \ n_j(\hat{x}_{j,t})}{\sum_{j=1}^N \ c_j(\hat{x}_{j,t})} < \lambda \vee \sum_{j=1}^{N} c_j(\hat{x}_{j,t}) > \beta\right) \leq \eta.$$

Our goal is the design of safe algorithms that minimize η. In particular, we are interested in safe algorithms guaranteeing that η increases sublinearly in (or independently of) T.

3 META-ALGORITHM

We provide the pseudo-code of our meta-algorithm in Algorithm 1. It solves the problem in Equations (1a)–(1c) in an online fashion. Algorithm 1 is based on three components: Gaussian Processes (GPs) [23] to model the parameters whose values are unknown, an *estimation subroutine* to generate estimates of the parameters from the GPs, and an *optimization subroutine* to solve the optimization problem given the estimates.

In particular, GPs are used to model the functions $n_j(\cdot)$ and $c_j(\cdot)$ describing the number of clicks and the costs, respectively. The employment of GPs to model these functions provides several advantages w.r.t. other regression techniques, such as the provision of a probability distribution over the possible values of the functions for every bid value $x \in X_j$ relying on a finite set of samples. GPs use the noisy realization of the number of clicks $\tilde{n}_{j,h}(\hat{x}_{j,h})$ collected from each subcampaign C_j for each past day $h \in \{1, \ldots, t-1\}$ to generate, for every bid $x \in X_j$, the estimates for the expected value $\hat{n}_{j,t-1}(x)$ and the standard deviation of the number of clicks $\hat{\sigma}^n_{j,t-1}(x)$. Analogously, using the noisy realizations of the cost function $\tilde{c}_{j,h}(\hat{x}_{j,h})$, with $h \in \{1, \ldots, t-1\}$, GPs generate, for every bid $x \in X_j$, the estimates for the expected value $\hat{c}_{j,t-1}(x)$ and the standard deviation of the costs $\hat{\sigma}^c_{j,t-1}(x)$. Details on the use of the GPs are provided by Rasmussen and Williams [23].

The estimation subroutine returns the vector $\boldsymbol{\mu}$ composed of the estimates generated from the GPs. In the following sections, we investigate two subroutines to compute $\boldsymbol{\mu}$. Then, the vector $\boldsymbol{\mu}$ is given as input to the optimization subroutine, called $\text{Opt}(\boldsymbol{\mu}, \lambda)$, that solves the problem stated in Equations (1a)–(1c) and returns the bid strategy $\left\{\hat{x}_{j,t}\right\}_{j=1}^N$ to play the next day t. Finally, once the strategy has been applied, the revenue $\sum_{j=1}^N v_j \ \tilde{n}_j(\hat{x}_{j,t})$ is obtained and the stochastic realization of the number of clicks $\tilde{n}_{j,t}(\hat{x}_{j,t})$ and costs $\tilde{c}_{j,t}(\hat{x}_{j,t})$ are observed and provided to the GPs to update the models used for the next day $t+1$. For the sake of presentation, we first present the $\text{Opt}(\boldsymbol{\mu}, \lambda)$ subroutine and, then, some estimation subroutines together with the theoretical guarantees provided by Algorithm 1 when these subroutines are adopted.

4 OPTIMIZATION SUBROUTINE

At first, we show that, even if all the values of the parameters of the optimization problem are known, the optimal solution cannot be approximated in polynomial time within any strictly positive factor (even depending on the size of the instance), unless P = NP. We reduce from SUBSET-SUM that is an NP-hard problem. Given a set S of integers $u_i \in \mathbb{N}^+$ and an integer $z \in \mathbb{N}^+$, SUBSET-SUM requires to decide whether there is a set $S^* \subseteq S$ with $\sum_{i \in S^*} u_i = z$.[4]

Theorem 1 (Inapproximability). *For any $\rho \in (0, 1]$, there is no polynomial-time algorithm returning a ρ-approximation to the problem in Equations* (1a)–(1c)*, unless* P = NP.

It is well known that SUBSET-SUM is a weakly NP-hard problem, admitting an exact algorithm whose running time is polynomial in the size of the problem and the magnitudes of the data involved rather than the base-two logarithm of their magnitudes. The same can be showed for our problem. Indeed, we can design a pseudo-polynomial-time algorithm to find the optimal solution in polynomial time w.r.t. the number of possible values of revenues and costs. In real-world settings, the values of revenue and cost are in limited ranges and rounded to the nearest cent, allowing the problem to be solved in a reasonable time. From now on, we assume for simplicity that the discretization of the ranges of the values of the daily cost Y and revenue R is evenly spaced.

The pseudo-code of the $\text{Opt}(\boldsymbol{\mu}, \lambda)$ subroutine, solving the problem in Equations (1a)–(1c) with a dynamic programming approach, is provided in Algorithm 2. It takes as input the set of the possible

[4] The proofs are deferred to the Supplementary Material.

Algorithm 2 Opt($\boldsymbol{\mu}, \lambda$) subroutine

Input: sets X_j of bid values, set Y of cumulative cost values, set R of revenue values, vector $\boldsymbol{\mu}$, ROI threshold λ

1: Initialize M empty matrix with dimension $|Y| \times |R|$
2: Initialize $\mathbf{x}^{y,r} = \mathbf{x}^{y,r}_{\text{next}} = [\ ]$, $\forall y \in Y, r \in R$
3: $S(y,r) = \bigcup \{x \in X_1 | \ \overline{c}_1(x) \leq y \wedge \underline{w}_1(x) \geq r\} \ \ \forall y \in Y, r \in R$
4: $\mathbf{x}^{y,r} = \arg\max_{x \in S} \overline{w}_1(x) \ \ \forall y \in Y, r \in R$
5: $M(y,r) = \max_{x \in S} \overline{w}_1(x) \ \ \forall y \in Y, r \in R$
6: **for** $j \in \{2, \dots, N\}$ **do**
7:     **for** $y \in Y$ **do**
8:         **for** $r \in R$ **do**
9:             Update $S(y,r)$ according to Equation (2)
10:             $\mathbf{x}^{y,r}_{\text{next}} = \arg\max_{\mathbf{s} \in S(y,r)} \sum_{i=1}^{j} \overline{w}_i(s_i)$
11:             $M(y,r) = \max_{\mathbf{s} \in S(y,r)} \sum_{i=1}^{j} \overline{w}_i(s_i)$
12:     $\mathbf{x}^{y,r} = \mathbf{x}^{y,r}_{\text{next}}$
13: Select (y^*, r^*) according to Equation (3)
14: **Output:** $\mathbf{x}^{y^*,r^*}$

bid values X_j for each subcampaign C_j, the set of the possible cumulative cost values Y such that $\max_{y \in Y} y = \beta$, the set of the possible revenue values R, a ROI threshold λ, and a vector of parameters characterizing the specific instance of the optimization problem:

$$\boldsymbol{\mu} := \left[\overline{w}_1(x_1), \dots, \overline{w}_N(x_{|X_N|}), \underline{w}_1(x_1), \dots, \underline{w}_N(x_{|X_N|}), \right. \\ \left. - \overline{c}_1(x_1), \dots, -\overline{c}_N(x_{|X_N|})\right],$$

where $w_j(x_j) := v_j \, n_j(x_j)$ denotes the revenue for a subcampaign C_j. We use $\overline{h}$ and $\underline{h}$ to denote potentially different estimated values of a generic function h used by the learning algorithms in the next sections. In particular, if the functions are known beforehand, then it holds $\overline{h} = \underline{h} = h$ for both $h = w_j$ and $h = c_j$. For the sake of clarity, $\overline{w}_j(x)$ is used in the objective function, while $\underline{w}_j(x)$ and $\overline{c}_j(x)$ are used in the constraints. At first, the subroutine initializes a matrix M in which it stores the optimal solution for each combination of values $y \in Y$ and $r \in R$, and initializes the vectors $\mathbf{x}^{y,r} = \mathbf{x}^{y,r}_{\text{next}} = [\ ]$, $\forall y \in Y, \forall r \in R$ (Lines 1–2). Then, the subroutine generates the set $S(y,r)$ of the bids for subcampaign C_1 (Line 3). More precisely, the set $S(y,r)$ contains only the bids x that induce the overall costs to be lower or equal than y and the overall revenue to be higher or equal than r. The bid in $S(y,r)$ that maximizes the revenue calculated with parameters $\overline{w}_j$ is included in the vector $\mathbf{x}^{y,r}$, while the corresponding revenue is stored in the matrix M. Then, the subroutine iterates over each subcampaign C_j, with $j \in \{2, \dots, N\}$, all the values $y \in Y$, and all the values $r \in R$ (Lines 9–11). At each iteration, for every pair (y,r), the subroutine stores in $\mathbf{x}^{y,r}$ the optimal set of bids for subcampaigns $C_1, \dots, C_j$ that maximizes the objective function, and stores the corresponding optimum value in $M(y,r)$. At every j-th iteration, the computation of the optimal bids is performed by evaluating a set of candidate solutions $S(y,r)$, computed as follows:

$$S(y,r) := \bigcup \Big\{\mathbf{s} = [\mathbf{x}^{y',r'}, x] \text{ s.t. } \ y' + \overline{c}_j(x) \leq y \ \wedge \\ r' + \underline{w}_j(x) \geq r \wedge x \in X_j \wedge y' \in Y \wedge r' \in R\Big\}. \tag{2}$$

This set is built by combining the optimal bids $\mathbf{x}^{y',r'}$ computed at the $(j-1)$-th iteration with one of the bids $x \in X_j$ available for the j-th subcampaign, such that these combinations satisfy the ROI and budget constraints. Then, the subroutine assigns the element of $S(y,r)$ that maximizes the revenue to $\mathbf{x}^{y,r}_{\text{next}}$ and the corresponding revenue to $M(y,r)$. At the end, the subroutine computes the optimal pair (y^*, r^*) as follows:

$$(y^*, r^*) = \Big\{y \in Y, r \in R \text{ s.t. } \frac{r}{y} \geq \lambda \ \wedge \\ M(y,r) \geq M(y',r'), \quad \forall y' \in Y, \forall r' \in R\Big\}, \tag{3}$$

as well as the corresponding set of bids $\mathbf{x}^{y^*,r^*}$, containing one bid for each subcampaign. We can state the following:

Theorem 2 (Optimality). *Subroutine Opt($\boldsymbol{\mu}, \lambda$) returns the optimal solution to the problem in Equations* (1a)–(1c) *when* $\overline{w}_j(x) = \underline{w}_j(x) = v_j \, n_j(x)$ *and* $\overline{c}_j(x) = c_j(x)$ *for each* $j \in \{1, \dots, N\}$ *and the values of revenues and costs are in* R *and* Y*, respectively.*

The asymptotic running time of the Opt procedure is:

$$\Theta\left(\sum_{j=1}^{N} |X_j| \, |Y|^2 \, |R|^2\right),$$

where $|X_j|$ is the cardinality of the set of bids X_j, since it has to cycle over all the subcampaigns and, for each one of them, to find the maximum bids and compute the values in the matrix $S(y,r)$. Moreover, the asymptotic space complexity of the Opt procedure is $\Theta(\max_{j=\{1,\dots,N\}} |X_j| \, |Y| \, |R|)$ since it has to store the values in the matrix $S(y,r)$ and perform a maximum operation over the possible bids $x \in X_j$.

5 ESTIMATION SUBROUTINE

Initially, we focus on the nature of our learning problem, and we show that no online learning algorithm can provide a sublinear pseudo-regret while guaranteeing safety.

Theorem 3 (Pseudo-regret/safety tradeoff). *For every* $\epsilon > 0$ *and time horizon* T*, there is no algorithm with pseudo-regret smaller than* $(1/2 - \epsilon)\,T$ *that violates (in expectation) the constraints less than* $(1/2 - \epsilon)\,T$ *times.*

Notice that, for the sake of simplicity, our proof is based on the violation of (budget) Constraint (1c), but its extension to the violation of (ROI) Constraint (1b) is direct. Since we cannot simultaneously guarantee sublinear regret and a sublinear number of violations of the constraints, we focus on algorithms guaranteeing only one property. In particular, in the following, we provide two algorithms, the first guaranteeing sublinear regret and the second guaranteeing a sublinear number of violations of the constraints. The results provided in the following hold under the assumption that n_j and c_j can be modeled as GPs.

The asymptotic running time of the GP estimation subroutine is $\Theta(\sum_{j=1}^{N} |X_j| \, t^2)$, where t is the number of samples (current round), and the asymptotic space complexity is $\Theta(Nt^2)$, i.e., the space required to store the Gram matrix. The dependence on the number of days t due to the GP update procedure can be reduced to linear using the recursive formula for the GP mean and variance computation (see Chowdhury and Gopalan [7] for details).

Guaranteeing Sublinear Pseudo-regret: GCB.. Accabi et al. [1] provide the GCB algorithm, a combinatorial bandit algorithm in which the reward is modeled by a single GP. In this work, we use a specific instance of the GCB in which multiple parameters are modeled by independent GPs. The details on how to properly set the values in the vector $\boldsymbol{\mu}$ as prescribed by GCB are described in the Supplementary Material. The result provided in Theorem 1 by [1] bounds the GCB pseudo-regret in terms of the maximum information gain of the GP modeling the number of clicks of subcampaign C_j, formally defined as:

$$\gamma_{j,t} := \frac{1}{2} \max_{(x_{j,1},\ldots,x_{j,t}), x_{j,h} \in X_j} \left| I_t + \frac{\Phi(x_{j,1},\ldots,x_{j,t})}{\sigma^2} \right|,$$

where I_t is the identity matrix of order t, $\Phi(x_{j,1},\ldots,x_{j,t})$ is the Gram matrix of the GP computed on the vector $(x_{j,1},\ldots,x_{j,t})$, and $\sigma \in \mathbb{R}^+$ is the noise standard deviation.

From the above results, we can state the following:

Theorem 4 (GCB pesudo-regret). *Given $\delta \in (0, 1)$, GCB applied to the problem in Equations* (1a)–(1c)*, with probability at least $1-\delta$, suffers from a pseudo-regret of:*

$$R_T(GCB) \leq \sqrt{\frac{16TN^3 b_t}{\ln(1+\sigma^2)} \sum_{j=1}^{N} \gamma_{j,T}},$$

where $b_t := 2\ln\left(\frac{\pi^2 NQTt^2}{3\delta}\right)$ is an uncertainty term used to guarantee the confidence level required by GCB, and $Q := \max_{j\in\{1,\ldots,N\}} |X_j|$ is the maximum number of bids in a subcampaign.

On the other hand, the GCB algorithm violates (in expectation) the constraints a linear number of times in T.

Theorem 5 (GCB safety). *Given $\delta \in (0, 1)$, GCB applied to the problem in Equations* (1a)–(1c) *is η-safe where $\eta \geq T - \frac{\delta}{2NQT}$ and, therefore, the number of constraints violations is linear in T.*[5]

Guaranteeing Safety: GCB$_{\mathsf{safe}}$. We propose GCB$_{\mathsf{safe}}$, a variant of GCB relying on different values to be used in the vector $\boldsymbol{\mu}$. More specifically, we employ optimistic estimates for the parameters used in the objective function and pessimistic estimates for the parameters used in the constraints. Formally, in GCB$_{\mathsf{safe}}$, we set:

$$\overline{w}_j(x) := v_j \left[\hat{n}_{j,t-1}(x) + \sqrt{b_{t-1}}\hat{\sigma}^n_{j,t-1}(x)\right],$$
$$\underline{w}_j(x) := v_j \left[\hat{n}_{j,t-1}(x) - \sqrt{b_{t-1}}\hat{\sigma}^n_{j,t-1}(x)\right],$$
$$\overline{c}_j(x) := \hat{c}_{j,t-1}(x) + \sqrt{b_{t-1}}\hat{\sigma}^c_{j,t-1}(x).$$

Furthermore, GCB$_{\mathsf{safe}}$ needs a default set of bids $\left\{x^{\mathrm{d}}_{j,t}\right\}_{j=1}^{N}$, that is known *a priori* to be feasible for the problem in Equations (1a)–(1c) with the actual values of the parameters.[6] The pseudo-code of GCB$_{\mathsf{safe}}$ is provided in Algorithm 1 with the above definition of the parameters of vector $\boldsymbol{\mu}$, except that it returns $\left\{\hat{x}_{j,t}\right\}_{j=1}^{N} = \left\{x^{\mathrm{d}}_{j,t}\right\}_{j=1}^{N}$ if the optimization problem does not admit any feasible solution with the current estimates. We can show the following:

[5]In the Supplementary Material, we also present Theorem 9 that provides results on the magnitude of the violation of GCB.

[6]A trivial default feasible bid allocation is $\left\{x^{\mathrm{d}}_{j,t} = 0\right\}_{j=1}^{N}$.

Theorem 6 (GCB$_{\mathsf{safe}}$ safety). *Given $\delta \in (0, 1)$, GCB$_{\mathsf{safe}}$ applied to the problem in Equations* (1a)–(1c) *is δ-safe and, therefore, the number of constraints violations is constant in T.*

The safety property comes at the cost that GCB$_{\mathsf{safe}}$ may suffer from a much larger pseudo-regret than GCB:

Theorem 7 (GCB$_{\mathsf{safe}}$ pseudo-regret). *Given $\delta \in (0, 1)$, GCB$_{\mathsf{safe}}$ applied to the problem in Equations* (1a)–(1c) *suffers from a pseudo-regret $R_t(GCB_{\mathsf{safe}}) = \Theta(T)$.*

Guaranteeing Sublinear Pseudo-regret and Safety with Tolerance: GCB$_{\mathsf{safe}}(\psi)$. We can show that, when a tolerance in the violation of the constraints is accepted, GCB$_{\mathsf{safe}}$ can be exploited to obtain a sublinear pseudo-regret. We focus on the case in which we *a priori* know that the budget constraint is not active at the optimal solution. Similar results can be derived both when we *a priori* know that the ROI constraint is not active and when we have no *a priori* information on which constraint is active, see the Supplementary Material; furthermore, the extension to the case in which the budget constraint is not uncertain as it is guaranteed by the platform is direct. Given an instance of the problem in Equations (1a)–(1c) that we call *original problem*, we build an *auxiliary problem* in which we slightly relax the ROI constraint, substituting λ with $\lambda - \psi$. We define GCB$_{\mathsf{safe}}(\psi)$ as GCB$_{\mathsf{safe}}$ applied to the auxiliary problem. By definition, GCB$_{\mathsf{safe}}(\psi)$, w.h.p., does not violate the ROI constraint of the original problem by more than the tolerance ψ.

Theorem 8 (GCB$_{\mathsf{safe}}(\psi)$ pseudo-regret and safety with tolerance). *When $\psi \geq 2\frac{\beta_{opt}+n_{\max}}{\beta^2_{opt}} \sum_{j=1}^{N} v_j \sqrt{2\ln\left(\frac{\pi^2 NQT^3}{3\delta'}\right)}\sigma$ and $\beta_{opt} < \beta \frac{\sum_{j=1}^{N} v_j}{\frac{N\ \beta_{opt}\psi}{\beta_{opt}+n_{\max}} + \sum_{j=1}^{N} v_j}$, where $\delta' \leq \delta$, β_{opt} is the spend at the optimal solution of the original problem, and $n_{\max} := \max_{j,x} n_j(x)$ is the maximum over the sub-campaigns and the admissible bids of the expected number of clicks, GCB$_{\mathsf{safe}}$ provides a pseudo-regret w.r.t. the optimal solution to the original problem of $O\left(\sqrt{T\sum_{j=1}^{N}\gamma_{j,T}}\right)$ with probability at least $1-\delta-\frac{\delta'}{QT^2}$, while being δ-safe w.r.t. the constraints of the auxiliary problem.*

This result states that, on the result provided in Theorem 1 can be circumvented on a subset of the possible instances of the optimization problem, if we allow a violation of at most ψ of the ROI constraint. In this case, GCB$_{\mathsf{safe}}(\psi)$ guarantees sublinear regret and a number of constraints violations that is constant in T.

Notice that the magnitude of the violation ψ increases linearly in the maximum number of clicks n_{max} and $\sum_{j=1}^{N} v_j$, that, in its turn, increases linearly with the number of sub-campaigns N. This suggests that in large instances this value may be large. However, in practice, the maximum number of clicks of a sub-campaign $n_{\max}$ is a sublinear function in the optimal budget b_{opt}, and usually it goes to a constant as the budget spent goes to infinity. Moreover, the number of sub-campaigns N usually depends on the budget, i.e., the choice of the budget is such that the budget is linear in the number of sub-campaigns. Therefore, the result is that b_{opt} is of the same order of $\sum_{j=1}^{N} v_j$. In conclusion, since n_{max} is sublinear in b_{opt} and $\sum_{j=1}^{N} v_j$ is of the order of b_opt, the final expression of ψ is sub-linear in b_opt.

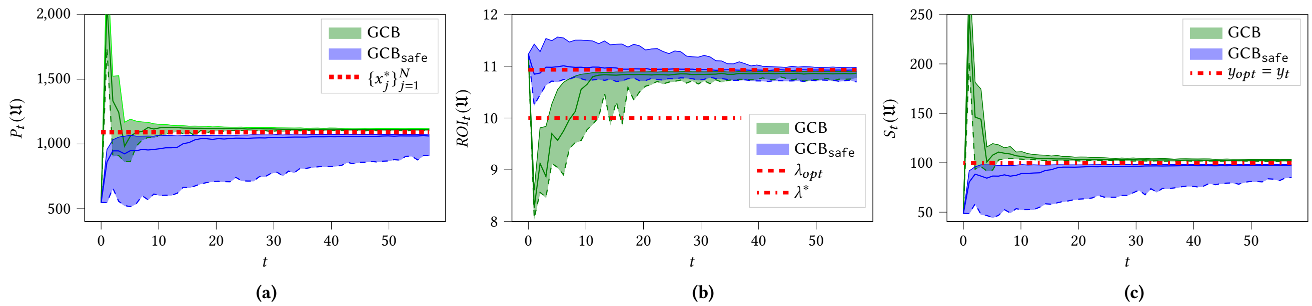


Figure 1: Results of Experiment #1: daily revenue (a), ROI (b), and spend (c) obtained by GCB and GCB_{safe}. Dash-dotted lines correspond to the optimum values for the revenue and ROI, while dashed lines correspond to the values of the ROI and budget constraints.

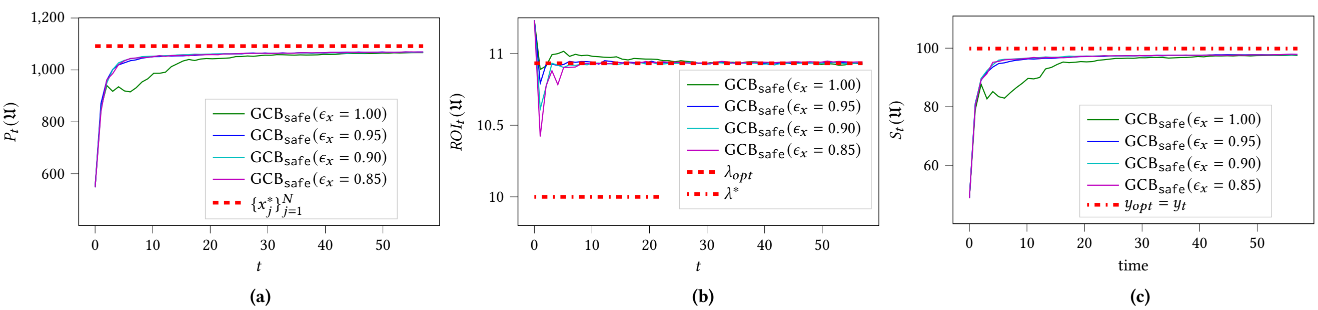


Figure 2: Results of Experiment #2: Median values of the daily revenue (a), ROI (b) and spend (c) obtained by GCB_{safe} with different values of ϵ_x.

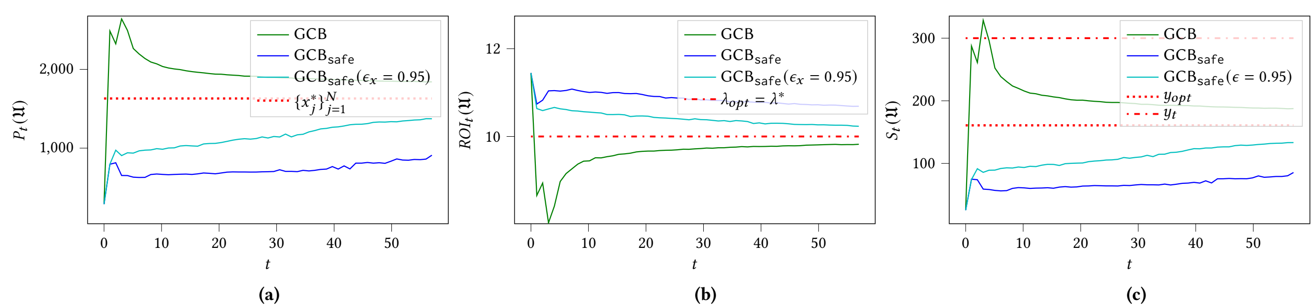


Figure 3: Results of Experiment #3: Median values of the daily revenue (a), ROI (b) and spend (c) obtained by GCB, GCB_{safe}, and $\text{GCB}_{\text{safe}}(\epsilon_x = 0.95)$.

6 EXPERIMENTAL EVALUATION

We compare the GCB algorithm with GCB_{safe} in synthetic settings, generated from real-world data, in terms of pseudo-regret and safety.

Experiment #1. We simulate $N = 5$ subcampaigns, with $|X_j| = 201$ bid values evenly spaced in $[0,\ 2]$, $|Y| = 101$ cost values evenly spaced in $[0,\ 100]$, and $|R| = 151$ revenue values evenly spaced in $[0,\ 1200]$. For a generic subcampaign C_j, at every t, the daily number of clicks is returned by function $\tilde{n}_j(x) := \theta_j(1 - e^{-x/\delta_j}) + \xi_j^n$ and the daily cost by function $\tilde{c}_j(x) = \alpha_j(1 - e^{-x/\gamma_j}) + \xi_j^c$, where $\theta_j \in \mathbb{R}^+$ and $\alpha_j \in \mathbb{R}^+$ represent the maximum achievable number of clicks and cost for subcampaign C_j in a single day, $\delta_j \in \mathbb{R}^+$ and $\gamma_j \in \mathbb{R}^+$ characterize how fast the two functions reach a saturation point and ξ_j^n and ξ_j^c are noise terms drawn from a $\mathcal{N}(0, 1)$ Gaussian distribution (these functions are customarily used in the advertising literature, *e.g.*, by Kong et al. [16]). The values used for the parameters of the above functions for the $N = 5$

subcampaigns have been estimated relying on a real-world dataset.[7] We assume a unitary value for each click, *i.e.,* $v_j = 1$ for each $j \in \{1, \ldots, N\}$. The values of the parameters of cost and revenue functions of the subcampaigns are specified in Table 1 reported in the Supplementary Material. We set a daily budget $\beta = 100$ for every t, $\lambda = 10$ in the ROI constraint, and a time horizon $T = 60$. The peculiarity of this setting is that, at the optimal solution, the budget constraint is active, while the ROI one is not (below, in Experiment #2, we study a setting in which the ROI constraint is active at the optimal solution).

For both GCB and $\mathrm{GCB}_{\mathsf{safe}}$, we use GPs with a squared exponential kernel of the form $k(x, x') := \sigma_f^2 \exp\left\{-\frac{(x-x')^2}{l}\right\}$ for each $x, x' \in X_j$, where the parameters $\sigma_f \in \mathbb{R}^+$ and $l \in \mathbb{R}^+$ are estimated from data, as suggested by Rasmussen and Williams [23]. The confidence for the algorithms is $\delta = 0.2$. We evaluate the algorithms in terms of:

- daily revenue: $P_t(\mathfrak{U}) := \sum_{j=1}^{N} v_j n_j(\hat{x}_{j,t})$;
- daily ROI: $ROI_t(\mathfrak{U}) := \frac{\sum_{j=1}^{N} v_j \ n_j(\hat{x}_{j,t})}{\sum_{j=1}^{N} \ c_j(\hat{x}_{j,t})}$;
- daily spend: $S_t(\mathfrak{U}) := \sum_{j=1}^{N} c_j(\hat{x}_{j,t})$.

We perform 100 independent runs for each algorithm.

Results. In Figure 1, for the daily revenue, ROI, and spend achieved by GCB and $\mathrm{GCB}_{\mathsf{safe}}$ at every t, we show the 50_{th} percentile (*i.e.*, the median) with solid lines and the 90_{th} and 10_{th} percentiles with dashed lines surrounding the semi-transparent area. While GCB achieves a larger revenue than $\mathrm{GCB}_{\mathsf{safe}}$, it violates the budget constraint over the entire time horizon and the ROI constraint in the first 7 days in more than 50% of the runs. This happens because, in the optimal solution, the ROI constraint is not active, while the budget constraint is. Conversely, $\mathrm{GCB}_{\mathsf{safe}}$ satisfies the budget and ROI constraints over the time horizon for more than 90% of the runs, and has a slower convergence to the optimum revenue. If we focus on the median revenue, $\mathrm{GCB}_{\mathsf{safe}}$ has a similar behaviour to that of GCB for $t > 15$. This makes $\mathrm{GCB}_{\mathsf{safe}}$ a good choice even in terms of overall revenue. However, it is worth to notice that, in the 10% of the runs, $\mathrm{GCB}_{\mathsf{safe}}$ does not converge to the optimal solution before the end of the learning period. These results confirm our theoretical analysis showing that limiting the exploration to safe regions might lead the algorithm to get large regret.

Experiment #2. We study a setting in which the ROI constraint is active at the optimal solution, *i.e.*, $\lambda = \lambda_{opt}$, while the budget constraint is not. This means that, at the optimal solution, the advertiser would have an extra budget to spend. However, such budget is not spent, otherwise the ROI constraint would be violated. The experimental setting is the same of Experiment #1, except that we set the budget constraint as $\beta = 300$. The optimal daily spend is $\beta_{opt} = 161$.

Results. In Figure 3, we show the median values of the daily revenue, the ROI, and the spend of GCB, $\mathrm{GCB}_{\mathsf{safe}}$, $\mathrm{GCB}_{\mathsf{safe}}(0.05)$. We notice that, even in this setting, GCB violates the ROI constraint for the entire time horizon, and the budget constraint in $t = 6$ and $t = 7$. However, it achieves a revenue larger than the optimum. On the other side, $\mathrm{GCB}_{\mathsf{safe}}$ always satisfies both the constraints, but it does not perform enough exploration to quickly converge to the optimal solution. We observe that it is sufficient to allow a tolerance in the ROI constraint violation by slightly perturbing the input value λ ($\psi = 0.05$, corresponding to a violation of the constraint by at most 5%) to make $\mathrm{GCB}_{\mathsf{safe}}$ capable of approaching the optimal solution while satisfying both constraints for every $t \in \{0, \ldots, T\}$. This suggests that, in real-world applications, $\mathrm{GCB}_{\mathsf{safe}}$ with a given tolerance represents an effective solution, providing guarantees on the violation of the constraints while returning high values of revenue. Such results are also confirmed by the additional experiments provided in the Supplementary Material.

7 CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a novel framework for Internet advertising campaigns. While previous works available in the literature focus only on the maximization of the revenue provided by the campaign, we introduce the concept of *safety* for the algorithms choosing the bid allocation each day. More specifically, we aim that the allocation satisfies, with high probability, some daily ROI and budget constraints fixed by the business units of the companies. The constraints are uncertain, as their parameters are not *a priori* known (some platforms do not allow the bidders to set daily budget constraint, while no platform allows the bidders to set daily constraints on ROI). Our goal is to maximize the revenue satisfying w.h.p. the uncertain constraints (a.k.a. safety). We model this setting as a constrained optimization problem, and we prove that such a problem is inapproximable within any strictly positive factor, unless $\mathsf{P} = \mathsf{NP}$, but it admits an exact pseudo-polynomial-time algorithm. Most interestingly, we prove that no online learning algorithm can provide sublinear pseudo-regret while guaranteeing a sublinear number of violations of the uncertain constraints. We show that the adaption of GCB suffers from a sublinear pseudo-regret, however, it may violate the constraints a linear number of times. Thus, we design $\mathrm{GCB}_{\mathsf{safe}}$, a novel algorithm that guarantees safety at the cost of a linear pseudo-regret. Remarkably, a simple adaptation of $\mathrm{GCB}_{\mathsf{safe}}$, namely $\mathrm{GCB}_{\mathsf{safe}}(\psi)$, guarantees a sublinear pseudo-regret and a safety with a fixed tolerance ψ. Finally, we evaluate the empirical performance of our algorithms on synthetically advertising problems generated from real-world data. These experiments show that $\mathrm{GCB}_{\mathsf{safe}}(\psi)$ provides good performance in terms of safety, while suffering from a small cumulative revenue w.r.t. GCB.

An interesting open research direction is the design of an algorithm which adopts constraints changing during the learning process, so as to identify the active constraint and relax those that are not active. Moreover, understanding the relationship between the relaxation of one of the constraints and the increase of the revenue constitutes an interesting line of research.

[7] The dataset is provided by AdsHotel (https://www.adshotel.com/), an Italian media agency working in the hotel booking market. The estimated values and the code used in the experiments are available at: https://github.com/oi-tech/safe_bid_opt.

REFERENCES

[1] G. M. Accabi, F. Trovò, A. Nuara, N. Gatti, and M. Restelli. 2018. When Gaussian Processes Meet Combinatorial Bandits: GCB. In *EWRL*.

[2] S. Amani, M. Alizadeh, and C. Thrampoulidis. 2020. Regret Bound for Safe Gaussian Process Bandit Optimization. In *L4DC*. 158–159.

[3] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. 2013. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 207–216.

[4] C. Borgs, J. Chayes, N. Immorlica, K. Jain, O. Etesami, and M. Mahdian. 2007. Dynamics of bid optimization in online advertisement auctions. In *WWW*. 531–540.

[5] X. Cao and K. J. Ray Liu. 2019. Online Convex Optimization With Time-Varying Constraints and Bandit Feedback. *IEEE T AUTOMAT CONTR* 64, 7 (2019), 2665–2680.

[6] W. Chen, Y. Wang, and Y. Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *ICML*. 151–159.

[7] S.R. Chowdhury and A. Gopalan. 2017. On kernelized multi-armed bandits. In *ICML*. 844–853.

[8] N. R. Devanur and S. M. Kakade. 2009. The price of truthfulness for pay-per-click auctions. In *ACM EC*. 99–106.

[9] W. Ding, T. Qin, X.-D. Zhang, and T.Y. Liu. 2013. Multi-Armed Bandit with Budget Constraint and Variable Costs. In *AAAI*. 232–238.

[10] J. Feldman, S. Muthukrishnan, M. Pal, and C. Stein. 2007. Budget optimization in search-based advertising auctions. In *ACM EC*. 40–49.

[11] N. Galichet, M. Sebag, and O. Teytaud. 2013. Exploration vs exploitation vs safety: Risk-aware multi-armed bandits. In *ACML*. 245–260.

[12] J. Garcia and F. Fernández. 2012. Safe exploration of state and action spaces in reinforcement learning. *J ARTIF INTELL RES* 45 (2012), 515–564.

[13] N. Golrezaei, I. Lobel, and R. Paes Leme. 2018. Auction design for ROI-constrained buyers. *Available at SSRN 3124929* (2018).

[14] A. Hans, D. Schneegaß, A. M. Schäfer, and S. Udluft. 2008. Safe exploration for reinforcement learning.. In *ESANN*. 143–148.

[15] IAB. 2020. Interactive Advertising Bureau (IAB) internet advertising revenue report, Full year 2019 results & Q1 2020 revenues. https://www.iab.com/wp-content/uploads/2020/05/FY19-IAB-Internet-Ad-Revenue-Report_Final.pdf. Online; accessed 4 January 2021.

[16] D. Kong, X. Fan, K. Shmakov, and J. Yang. 2018. A Combinational Optimization Approach for Advertising Budget Allocation. In *WWW*. 53–54.

[17] S. Mannor, J. N. Tsitsiklis, and J. Y. Yu. 2009. Online Learning with Sample Path Constraints. *J MACH LEARN RES* 10 (2009), 569–590.

[18] A. Moradipari, C. Thrampoulidis, and M. Alizadeh. 2020. Stage-wise Conservative Linear Bandits. In *NeurIPS*.

[19] A. Nuara, N. Sosio, F. Trovò, M. C. Zaccardi, N. Gatti, and M. Restelli. 2019. Dealing with Interdependencies and Uncertainty in Multi-Channel Advertising Campaigns Optimization. In *WWW*. 1376–1386.

[20] A. Nuara, F. Trovò, N. Gatti, and M. Restelli. 2018. A Combinatorial-Bandit Algorithm for the Online Joint Bid/Budget Optimization of Pay-per-Click Advertising Campaigns. In *AAAI*. 2379–2386.

[21] A. Nuara, F. Trovò, N. Gatti, and M. Restelli. 2020. Online Joint Bid/Daily Budget Optimization of Internet Advertising Campaigns. *CoRR* abs/2003.01452 (2020). arXiv:2003.01452 https://arxiv.org/abs/2003.01452

[22] M. Pirotta, M. Restelli, A. Pecorino, and D. Calandriello. 2013. Safe policy iteration. In *ICML*. 307–315.

[23] C. E. Rasmussen and C. K. Williams. 2006. *Gaussian processes for machine learning*. Vol. 1. MIT Press.

[24] N. Srinivas, A. Krause, M. Seeger, and S. M. Kakade. 2010. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *ICML*. 1015–1022.

[25] Y. Sui, A. Gotovos, J. Burdick, and A. Krause. 2015. Safe exploration for optimization with Gaussian processes. In *ICML*. 997–1005.

[26] B. K. Szymanski and J. Lee. 2006. Impact of roi on bidding and revenue in sponsored search advertisement auctions. In *Workshop on Sponsored Search Auctions*, Vol. 1.

[27] S. Thomaidou, K. Liakopoulos, and M. Vazirgiannis. 2014. Toward an integrated framework for automated development and optimization of online advertising campaigns. *INTELL DATA ANAL* 18, 6 (2014), 1199–1227.

[28] F. Trovò, S. Paladino, M. Restelli, and N. Gatti. 2016. Budgeted Multi-Armed Bandit in Continuous Action Space. In *ECAI*. 560–568.

[29] V. V. Vazirani, N. Nisan, T. Roughgarden, and E. Tardos. 2007. *Algorithmic Game Theory*. Cambridge University Press.

[30] W. Zhang, Y. Zhang, B. Gao, Y. Yu, X. Yuan, and T.-Y. Liu. 2012. Joint optimization of bid and budget allocation in sponsored search. In *SIGKDD*. 1177–1185.

A SUPPLEMENTARY MATERIAL FOR THE PAPER "SAFE ONLINE BID OPTIMIZATION WITH UNCERTAIN RETURN-ON-INVESTMENT AND BUDGET CONSTRAINTS"

A.1 Optimization Subroutine Analysis

THEOREM 1 (INAPPROXIMABILITY). *For any $\rho \in (0, 1]$, there is no polynomial-time algorithm returning a ρ-approximation to the problem in Equations* (1a)–(1c)*, unless* P = NP.

PROOF. We restrict to the instances of SUBSET-SUM such that $z \le \sum_{i\in S} u_i$. Solving these instances is trivially NP-hard, as any instance with $z > \sum_{i\in S} u_i$ is not satisfiable, and we can decide it in polynomial time. Given an instance of SUBSET-SUM, let $\ell = \frac{\sum_{i\in S} u_i+1}{\rho}$. Let us notice that, the lower the degree of approximation we aim, the larger the value of ℓ. For instance, when study the problem of computing an exact solution, we set $\rho = 1$ and therefore $\ell = \sum_{i\in S} u_i + 1$, whereas, when we require a 1/2-approximation, we set $\rho = 1/2$ and therefore $\ell = 2(\sum_{i\in S} u_i + 1)$. We have $|S| + 1$ subcampaigns, each denoted with C_j. The available bids belong to $\{0, 1\}$ for every subcampaign C_j. The parameters of the subcampaigns are set as follows:

- subcampaign C_0: we set $v_0 = 1$, and

$$c_0(x) = \begin{cases} 2\ell + z & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}, \qquad n_0(x) = \begin{cases} \ell & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases};$$

- subcampaign C_j for every $j \in S$: we set $v_j = 1$, and

$$c_j(x) = \begin{cases} u_i & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}, \qquad n_j(x) = \begin{cases} u_i & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}.$$

We set the daily budget $\beta = 2(z + \ell)$ and the ROI limit $\lambda = \frac{1}{2}$.[8]

We show that, if a SUBSET-SUM instance is satisfiable, then the corresponding instance of our problem admits a solution with a revenue larger than ℓ, while, if a SUBSET-SUM instance is not satisfiable, the maximum revenue in the corresponding instance of our problem is at most $\rho\,\ell - 1$. Thus, the application of any polynomial-time ρ-approximation algorithm to instances of our problem generated from instances of SUBSET-SUM as described above would return a solution whose value is not smaller than $\rho\,\ell$ when the SUBSET-SUM instance is satisfiable and it is not larger than $\rho\,\ell - 1$ when the SUBSET-SUM instance is not satisfiable. As a result, whenever such an algorithm returns a solution with a value that is not smaller than $\rho\,\ell$, we can decide that the corresponding SUBSET-SUM instance is satisfiable. Analogously, whenever such an algorithm returns a solution with a value that is in the range $[\rho(\rho\,\ell - 1), \rho\,\ell - 1]$, we can decide that the corresponding SUBSET-SUM instance is not satisfiable. Let us notice that the range $[\rho(\rho\,\ell - 1), \rho\,\ell - 1]$ is well defined for every $\rho \in (0, 1]$, as, by construction, $\rho\,\ell = \sum_{i\in S} u_i + 1 \ge 1$ and therefore $\rho\,\ell - 1 \ge \rho(\rho\,\ell - 1)$. Hence, such an algorithm would decide in polynomial time whether or not a SUBSET-SUM instance is satisfiable, but this is not possible unless P = NP. Since this holds for every $\rho \in (0, 1]$, then no ρ-approximation to our problem is allowed in polynomial time unless P = NP.

If. Suppose SUBSET-SUM is satisfied by the set $S^* \subseteq S$ and that the solution assigns $x_i = 1$ if $i \in S^*$ and $x_i = 0$ otherwise, and it assigns $x_0 = 1$. The total revenue is $\ell + z \ge \ell$ and the constraints are satisfied. In particular, the sum of the costs is $2\ell + z + z = 2(\ell + z)$, while ROI $= \frac{\ell+z}{2\ell+2z} = \frac{1}{2}$.

Only if. Assume by contradiction that the instance of our problem admits a solution with a revenue strictly larger than $\rho\,\ell - 1$ and that SUBSET-SUM is not satisfiable. Then, it is easy to see that we need $x_0 = 1$ for campaign C_0 as the maximum achievable revenue is $\sum_{i\in S} u_i = \rho\,\ell - 1$ when $x_0 = 0$. Thus, since $x_0 = 1$, the budget constraint forces $\sum_{i\in S: x_i=1} c_i(x_i) \le z$, thus implying $\sum_{i\in S: x_i=1} u_i \le z$. By the satisfaction of the ROI constraint, *i.e.*, $\frac{\sum_{i\in S: x_i=1} u_i + l}{\sum_{i\in S: x_i=1} u_i + 2l + z} \ge \frac{1}{2}$, it must hold $\sum_{i\in S: x_i=1} u_i \ge z$. Therefore, the set $S^* = \{i \in S : x_i = 1\}$ is a solution to SUBSET-SUM, thus reaching a contradiction. This concludes the proof. □

THEOREM 2 (OPTIMALITY). *Subroutine Opt($\boldsymbol{\mu}, \lambda$) returns the optimal solution to the problem in Equations* (1a)–(1c) *when $\overline{w}_j(x) = \underline{w}_j(x) = v_j\, n_j(x)$ and $\overline{c}_j(x) = c_j(x)$ for each $j \in \{1, \ldots, N\}$ and the values of revenues and costs are in R and Y, respectively.*

PROOF. Since all the possible values for the revenues and costs are taken into account in the subroutine, the elements in $S(y, r)$ satisfy the two inequalities in Equation (2) with the equal sign. Therefore, all the elements in $S(y, r)$ would contribute to the computation of the final value of the ROI and budget constraints, *i.e.*, the ones after evaluating all the N subcampaigns, with the same values for revenue and costs, being their overall revenue equal to r and their overall cost equal to y. Notice that Constraint (1c) is satisfied as long as it holds $\max(Y) = \beta$. The maximum operator in Line 11 excludes only solutions with the same costs and a lower revenue, therefore, the subroutine excludes only solutions that would never be optimal (and, for this reason, said dominated). The same reasoning holds also for the subcampaign C_1 analysed by the algorithm. Finally, after all the dominated allocations have been discarded, the solution is selected by Equation (3), *i.e.*, among all the solutions satisfying the ROI constraints the one with the largest revenue is selected. □

[8] For the ease of exposition, the proof uses simple instances. The adoption of simple cases is crucial to identify the most basic settings in which the problem is hard, and it is customary in the theory literature. Let us notice that it is possible to prove the theorem using instances that satisfy real-world assumptions. For example, we can build a reduction in which the costs are smaller than the values, *i.e.*, $c_i(x) < n_i(x)v_i$. In particular, the reduction holds even if we set $c_0(1) = \epsilon(2l + z)$, $c_j(1) = \epsilon u_i$, $\beta = 2\epsilon(z + l)$, and $\lambda = 1/(2\epsilon)$ for an arbitrary small ϵ.

In what follows, we provide an impossibility result for the optimization problem in Equations (1a)–(1c). For the sake of simplicity, our proof is based on the violation of (budget) Constraint (1c), but its extension to the violation of (ROI) Constraint (1b) is direct.

Theorem 3 (Pseudo-regret/safety tradeoff). *For every $\epsilon > 0$ and time horizon T, there is no algorithm with pseudo-regret smaller than $(1/2 - \epsilon)\,T$ that violates (in expectation) the constraints less than $(1/2 - \epsilon)\,T$ times.*

Proof. Initially, we show that an algorithm satisfying the two conditions of the theorem can be used to distinguish between $\mathcal{N}(1, 1)$ and $\mathcal{N}(1 + \delta, 1)$ with an arbitrarily large probability using a number of samples independent from δ. Consider two instances of the bid optimization problem defined as follows. Both instances have a single subcampaign with $x \in \{0, 1\}$, $c(0) = 0$, $r(0) = 0$, $r(1) = 1$, $\beta = 1$, and $\lambda = 0$. The first instance has cost $c^1(1) = \mathcal{N}(1, 1)$, while the second one has $c^2(1) = \mathcal{N}(1 + \delta, 1)$. With the first instance, the algorithm must choose $x = 1$ at least $T(1/2 + \epsilon)$ times in expectation, otherwise the pseudo-regret would be strictly greater than $T(1/2 - \epsilon)$, while, with the second instance, the algorithm must choose $x = 1$ at most than $T(1/2 - \epsilon)$ times in expectation, otherwise the constraint on the budget would be violated strictly more than $T(1/2 - \epsilon)$ times. Standard concentration inequalities imply that, for each $\gamma > 0$, there exists a $n(\epsilon, \gamma)$ such that, given $n(\epsilon, \gamma)$ runs of the learning algorithm, with the first instance the algorithm plays $x = 1$ strictly more than $Tn(\epsilon, \gamma)/2$ times with probability at least $1 - \gamma$, while with the second instance it is played strictly less than $Tn(\epsilon, \gamma)/2$ times with probability at least $1 - \gamma$. This entails that the learning algorithm can distinguish with arbitrarily large success probability (independent of δ) between the two instances using (at most) $n(\epsilon, \gamma)T$ samples from one of the normal distributions.

However, the Kullback-Leibler divergence between the two normal distributions is $KL(\mathcal{N}(1, 1), \mathcal{N}(1 + \delta, 1)) = \delta^2/2$ and each algorithm needs at least $\Omega(1/\delta^2)$ samples to distinguish between the two distributions with arbitrarily large probability. Since δ can be arbitrarily small, we have a contradiction. Thus, such an algorithm cannot exist. This concludes the proof.[9] □

A.2 Applying GCB to the Bid Optimization Problem

In what follows we provide the full description of the GCB algorithm applied to the problem of advertisement and state the assumptions required to provide theoretical guarantees on the regret.

To guarantee that GCB provides a sublinear pseudo-regret, we need that a few assumptions are satisfied. More specifically, we need a *monotonicity property*, stating that the value of the objective function increases as the values of the elements in $\boldsymbol{\mu}$ increase and a *Lipschitz continuity* assumption between the parameter vector $\boldsymbol{\mu}$ and the value returned by the objective function in Equation (1a). Formally:

Assumption 1 (Monotonicity)**.** *The expected reward $r_{\boldsymbol{\mu}}(S) := \sum_{j=1}^{N} v_j\; n_j(x_{j,t})$, where S is the bid allocation, is monotonically non decreasing in $\boldsymbol{\mu}$, i.e., given $\boldsymbol{\mu}$, $\boldsymbol{\eta}$ s.t. $\mu_i \le \eta_i$ for each i, we have $r_{\boldsymbol{\mu}}(S) \le r_{\boldsymbol{\eta}}(S)$ for each S.*

and:

Assumption 2 (Lipschitz continuity)**.** *The expected reward $r_{\boldsymbol{\mu}}(S)$ is Lipschitz continuous in the infinite norm w.r.t. the expected payoff vector $\boldsymbol{\mu}$, with Lipschitz constant $\Lambda > 0$. Formally, for each $\boldsymbol{\mu}, \boldsymbol{\eta}$ we have $|r_{\boldsymbol{\mu}}(S) - r_{\boldsymbol{\eta}}(S)| \le \Lambda ||\boldsymbol{\mu} - \boldsymbol{\eta}||_\infty$, where the infinite norm of a payoff vector is $||\boldsymbol{\mu}||_\infty := \max_i |\mu_i|$.*

While it is easy to show that Lipschitz continuity holds with constant $\Lambda = N$ (number of subcampaigns), the monotonicity property holds by definition of $\boldsymbol{\mu}$, as the increase of a value of $\overline{w}_j(x)$ would increase the value of the objective function, and the increase of the values of $\underline{w}_j(x)$ or $\overline{c}_j(x)$ would enlarge the feasibility region of the problem, thus not excluding optimal solutions.

The GCB algorithms is presented in Algorithm 3. It uses two sets of GPs to estimate the number of clicks and the costs functions, one for each subcampaigns C_j with $j \in \{1, \dots, N\}$. Then, the estimated payoffs for each arm $x_{j,t}$ are fed to the $\mathrm{Opt}(\boldsymbol{\mu}, \lambda)$procedure which chooses the super-arm S_t to play at round t. The algorithm requires as input the set of bids X_j for each subcampaign, a prior for each one of the GPs specified by the mean function $\hat{n}_{j,0}(\cdot)$ and the standard deviation function $\hat{\sigma}^n_{j,0}(\cdot)$ for the number of clicks and the mean function $\hat{c}_{j,0}(\cdot)$ and the standard deviation function $\hat{\sigma}^c_{j,0}(\cdot)$ for the costs. At round t, the algorithm computes estimates for the expected payoff for each bid $x \in X_j$. The algorithm relies on the observations provided by the advertisement process up to time $t - 1$ by means of the values of the gram matrix $K_{i,t}$ of the number of clicks and $H_{i,t}$ of the costs. It also requires to compute the vector of the covariance between the analysed bid x and each bid seen up to now $\tilde{x}_{j,t}$, formally $k_{j,t-1} := [k_j(\tilde{x}_{j,1}, x), \dots k_j(\tilde{x}_{j,t-1}, x)]$ and $h_{j,t-1} := [h_j(\tilde{x}_{j,1}, x), \dots h_j(\tilde{x}_{j,t-1}, x)]$, where $k_j(\cdot, \cdot)$ and $h_j(\cdot, \cdot)$ are the kernel functions for the number of clicks and the costs. Such a model provides a probability distribution for each expected payoff, which is not directly employable in the approximation oracle, that, instead, needs a single value per expected payoff vector. We cope with this issue we rely on upper an upper confidence bounds $\boldsymbol{\mu}$ over the considered quantities:

$$\overline{w}_j(x) = \underline{w}_j(x) := v_j \left[\hat{n}_{j,t-1}(x) + \sqrt{b_{t-1}}\,\hat{\sigma}^n_{j,t-1}(x)\right], \tag{4}$$

$$\overline{c}_j(x) := \hat{c}_{j,t-1}(x) - \sqrt{b_{t-1}}\,\hat{\sigma}^c_{j,t-1}(x), \tag{5}$$

[9] Notice The theorem can be modified to hold even with instances that satisfy real-world assumptions, *e.g.*, with costs much smaller than the budget. Indeed, we can apply the same reduction in which the costs are arbitrary, *e.g.*, $c(0) = c(1) = q$ with an arbitrary small q and $\beta = 1$, while the utilities are $r(0) = 0$, $r(1) = \mathcal{N}(1, 1)$ or $r(1) = \mathcal{N}(1 - \delta, 1)$, and the ROI limit is $\lambda = 1/q$.

Algorithm 3 GCB Algorithm

Input: Set of bids X_j, noise variance σ^2, GP Prior distributions $\hat{n}_{j,0}$, $\hat{\sigma}^n_{j,0}$, $\hat{c}_{j,0}$, and $\hat{\sigma}^c_{j,0}$ for all $i \in \{1, \dots, N\}$

for $t \in \{1, \dots, T\}$ **do**
  for $j \in \{1, \dots, N\}$ **do**
    for $x \in X_j$ **do**
      Compute estimates $\hat{n}_{j,t-1}(x) := k_{j,t-1}(x)^\top (K_{j,t-1} + \sigma^2 I)^{-1} k_{j,t-1}(x)$
      Compute estimates $\hat{\sigma}^n_{j,t-1}(x) := k_j(x,x) - k^\top_{j,t-1}(K_{j,t-1} + \sigma^2 I)^{-1} k_{j,t-1}(x)$
      Compute estimates $\hat{c}_{j,t-1}(x) := h_{j,t-1}(x)^\top (H_{j,t-1} + \sigma^2 I)^{-1} h_{j,t-1}(x)$
      Compute estimates $\hat{\sigma}^c_{j,t-1}(x) := h_j(x,x) - h^\top_{j,t-1}(H_{j,t-1} + \sigma^2 I)^{-1} h_{j,t-1}(x)$
  Compute $\boldsymbol{\mu}$ using the GPs estimates
  Run the $\mathrm{Opt}(\boldsymbol{\mu}, \lambda)$ procedure to get a solution $\left\{\hat{x}_{j,t}\right\}_{j=1}^N$
  Set the prescribed allocation during day t
  Get revenue $\sum_{j=1}^N v_j \ \tilde{n}_j(\hat{x}_{j,t})$
  Update the GPs using the new information $\tilde{n}_{j,t}(\hat{x}_{j,t})$ and $\tilde{c}_{j,t}(\hat{x}_{j,t})$

where $b_t := 2 \ln \left(\frac{\pi^2 NQTt^2}{3\delta} \right)$ is an uncertainty term used to guarantee the confidence level required by GCB. Note that, given $\delta \in (0, 1)$, $\overline{w}_j(x)$ and $\underline{w}_j(x)$ are statistical upper bounds for the actual values $n_j(x)$ and that $\overline{c}_j(x)$ are statistical lower bounds for the actual values $c_j(x)$ holding for all $x \in X_j$ and for all $j \in \{1, \dots, N\}$ with probability at least $1 - \delta$ for $t \in \{1, \dots, T\}$.

For the sake of simplicity, we assume that the values of the bounds correspond to values in R and Y, respectively. If the bound values for $\overline{w}_j(x)$ are not in the set R, we need to round them up to the nearest value belonging to R. Instead, if $\underline{c}_j(x)$ are not in the set Y, a rounding down should be performed to the nearest value in Y.

Theorem 4 (GCB pesudo-regret). *Given $\delta \in (0,\ 1)$, GCB applied to the problem in Equations* (1a)–(1c)*, with probability at least* $1 - \delta$*, suffers from a pseudo-regret of:*

$$R_T(GCB\,) \leq \sqrt{\frac{16TN^3 b_t}{\ln(1+\sigma^2)} \sum_{j=1}^{N} \gamma_{j,T}}\ ,$$

where $b_t := 2 \ln \left(\frac{\pi^2 NQTt^2}{3\delta} \right)$ is an uncertainty term used to guarantee the confidence level required by GCB, and $Q := \max_{j \in \{1,\dots,N\}} |X_j|$ is the maximum number of bids in a subcampaign.

Proof. The bounds in Equations (4) and (5) guarantee that the probability that there is at least a triple (j, x, t) with $j \in N$, $x \in X_j$, $t \in \{1, \dots, T\}$ such that the actual value of $v_j n_j(x)$ is larger than the upper bound $\overline{w}_{j,t-1}(x) = \underline{w}_{j,t-1}(x)$ or the actual value of $c_j(x)$ is smaller than the lower bound $\overline{c}_{j,t-1}(x)$ is less than $\delta/2$ (see Accabi et al. [1] for details). This implies, using a union bound, that the values in $\boldsymbol{\mu}$ used in the oracle $\mathrm{Opt}(\boldsymbol{\mu}, \lambda)$ are statistical (optimistical) bounds for the true values with probability at least $1 - \delta$, as required by GCB. Then, the proof follows by applying Theorem 1 by Accabi et al. [1] to our setting, using that $\mathrm{Opt}(\boldsymbol{\mu}, \lambda)$ subroutine is an (α, β)-approximation algorithm with $\alpha = 1$ and $\beta = 1$ (see Chen et al. [6] for a formal definition). □

Theorem 5 (GCB safety). *Given $\delta \in (0,\ 1)$, GCB applied to the problem in Equations* (1a)–(1c) *is η-safe where $\eta \geq T - \frac{\delta}{2NQT}$ and, therefore, the number of constraints violations is linear in T.*[10]

Proof. Let us focus on a specific day t. Consider the case in which Constraints (1b) and (1c) are active, and, therefore, the left side equals the right side: $\sum_{j=1}^N \underline{w}_j(x_{j,t}) - \lambda \sum_{j=1}^N \ \overline{c}_j(x_{j,t}) = 0$ and $\sum_{j=1}^N \overline{c}_j(x_{j,t}) = \beta$. For the sake of simplicity we focus on the costs $\overline{c}_j(x_{j,t})$, but similar arguments also applies to the revenues $\underline{w}_j(x_{j,t})$. A necessary condition for which the two constraints are valid also for the real revenue and costs is that for at least one of the costs it holds $c_j(x_{j,t}) \leq \overline{c}_j(x_{j,t})$. Indeed, if the opposite holds, *i.e.*, $\overline{c}_j(x_{j,t}) < c_j(x_{j,t})$ for each $j \in \{1, \dots, N\}$ and $x_{j,t} \in X_j$, the budget constraint would be violated by the allocation since $\sum_{j=1}^N c_j(x_{j,t}) > \sum_{j=1}^N \overline{c}_j(x_{j,t}) = \beta$. Since the event $c_j(x_{j,t}) \leq \overline{c}_j(x_{j,t})$ occurs with probability at most $\frac{3\delta}{\pi^2 NQTt^2}$, over the $t \in \mathbb{N}$, formally:

$$\mathbb{P}\left(\frac{\sum_{j=1}^N v_j \ n_j(\hat{x}_{j,t})}{\sum_{j=1}^N \ c_j(\hat{x}_{j,t})} < \lambda \vee \sum_{j=1}^{N} c_j(\hat{x}_{j,t}) > \beta \right) \geq 1 - \frac{3\delta}{\pi^2 NQTt^2}.$$

[10] In the Supplementary Material, we also present Theorem 9 that provides results on the magnitude of the violation of GCB.

Finally, summing over the time horizon T the probability that the constraints are not violated is at most $\frac{\delta}{2NQT}$, formally:

$$\sum_{t=1}^{T} \mathbb{P}\left(\frac{\sum_{j=1}^{N} v_j \, n_j(\hat{x}_{j,t})}{\sum_{j=1}^{N} \, c_j(\hat{x}_{j,t})} < \lambda \vee \sum_{j=1}^{N} c_j(\hat{x}_{j,t}) > \beta \right) \geq T - \frac{\delta}{2NQT}.$$

□

In the following theorem we want to show that the cumulated violation by GCB of at least one of the constraints algorithm is bounded. The following results assume that each subcampaign have a minimum cost per day $c_{\min} > 0$, a maximum cost $c_{\max}$, and a maximum number of clicks $n_{\max} := \max_{j \in \{1,\dots,N\}, x \in X} n_j(x)$.

Theorem 9 (GCB cumulated violation). *The cumulated violation of the two constraints provided by the GCB algorithm satisfies:*

- $\sum_{t=1}^{T} \sum_{j=1}^{N} c_j(x_{j,t}) - T\,\beta \leq O\left(\sqrt{T \sum_{j=1}^{N} \gamma_{j,T}^{c}}\right)$,
- $T\lambda - \sum_{t=1}^{T} \frac{\sum_{j=1}^{N} v_j n_j(x_{j,t})}{\sum_{j=1}^{N} c_j(x_{j,t})} \leq O\left(\sqrt{T \sum_{j=1}^{N} (\gamma_{j,t} + \gamma_{j,t}^{c})}\right)$,

where $\gamma_{j,t}^{c}$ is the maximum information gain of the GPs modeling the costs of j-th subcampaign after t samples.

Proof. We analyse the violation of the ROI constraint vr_t at a specific day t and the one of the budget constraint vb_t.

Focusing on the budget constraint, we have:

$$vb_j = \sum_{j=1}^{N} c_j(x_{j,t}) - y \leq \sum_{j=1}^{N} (\hat{c}_j(x_{j,t}) + \sqrt{b_{t-1}} \hat{\sigma}_{j,t-1}^{c}(x_{j,t})) - \beta \tag{6}$$

$$= \underbrace{\sum_{j=1}^{N} (\hat{c}_j(x_{j,t}) - \sqrt{b_{t-1}} \hat{\sigma}_{j,t-1}^{c}(x_{j,t})) - \beta}_{\leq 0} + 2 \sum_{j=1}^{N} \sqrt{b_{t-1}} \hat{\sigma}_{j,t-1}^{c}(x_{j,t}) \tag{7}$$

$$\leq 2 \sum_{j=1}^{N} \sqrt{b_{t-1}} \hat{\sigma}_{j,t-1}^{c}(x_{j,t}), \tag{8}$$

where the inequality in Equation (7) holds from the fact that the solution selected by GCB has to satisfy the budget constraint. Define $\overline{n}_j(x_{j,t}) := \hat{n}_j(x_{j,t}) + \sqrt{b_{t-1}} \hat{\sigma}_j^{n}(x_{j,t})$. Notice that the previous bound holds w.p. at least $1 - \delta$ due to the fact that this is the probability for which the bounds on the number of clicks and the costs hold.

Since we have $\lambda \leq \frac{\sum_{j=1}^{N} v_j \overline{n}_j(x_{j,t})}{\sum_{j=1}^{N} \overline{c}_j(x_{j,t})}$:

$$vr_t = \lambda - \frac{\sum_{j=1}^{N} v_j n_j(x_{j,t})}{\sum_{j=1}^{N} c_j(x_{j,t})} \leq \frac{\sum_{j=1}^{N} v_j \overline{n}_j(x_{j,t})}{\sum_{j=1}^{N} \overline{c}_j(x_{j,t})} - \frac{\sum_{j=1}^{N} v_j n_j(x_{j,t})}{\sum_{j=1}^{N} c_j(x_{j,t})} \tag{9}$$

$$\leq \frac{\sum_{j=1}^{N} c_j(x_{j,t}) \sum_{j=1}^{N} v_j \overline{n}_j(x_{j,t}) - \sum_{j=1}^{N} \overline{c}_j(x_{j,t}) \sum_{j=1}^{N} v_j n_j(x_{j,t})}{\sum_{j=1}^{N} c_j(x_{j,t}) \sum_{j=1}^{N} \overline{c}_j(x_{j,t})} \tag{10}$$

$$\begin{aligned} \leq & \frac{1}{N^2 c_{\min}(c_{\min} - \sqrt{b_T}\sigma)} \left(\sum_{j=1}^{N} c_j(x_{j,t}) \sum_{j=1}^{N} v_j \overline{n}_j(x_{j,t}) - \sum_{j=1}^{N} c_j(x_{j,t}) \sum_{j=1}^{N} v_j n_j(x_{j,t}) \right. \\ & \left. + \sum_{j=1}^{N} c_j(x_{j,t}) \sum_{j=1}^{N} v_j n_j(x_{j,t}) - \sum_{j=1}^{N} \overline{c}_j(x_{j,t}) \sum_{j=1}^{N} v_j n_j(x_{j,t}) \right) \end{aligned} \tag{11}$$

$$\begin{aligned} \leq & \frac{1}{N^2 c_{\min}(c_{\min} - \sqrt{b_T}\sigma)} \left[\sum_{j=1}^{N} c_j(x_{j,t}) \left(\sum_{j=1}^{N} v_j \overline{n}_j(x_{j,t}) - \sum_{j=1}^{N} v_j n_j(x_{j,t}) \right) \right. \\ & \left. + \sum_{j=1}^{N} v_j n_j(x_{j,t}) \left(\sum_{j=1}^{N} c_j(x_{j,t}) - \sum_{j=1}^{N} \overline{c}_j(x_{j,t}) \right) \right] \end{aligned} \tag{12}$$

$$\leq \frac{N c_{\max} v_{\max} 2 \sum_{j=1}^{N} \sqrt{b_{t-1}} \hat{\sigma}_j^{n}(x_{j,t}) + N n_{\max} v_{\max} 2 \sum_{j=1}^{N} \sqrt{b_{t-1}} \hat{\sigma}_j^{c}(x_{j,t})}{N^2 c_{\min}(c_{\min} - \sqrt{b_T}\sigma)} \tag{13}$$

$$= \frac{2c_{\max} v_{\max} \sum_{j=1}^{N} \sqrt{b_{t-1}} \hat{\sigma}_j^n(x_{j,t}) + 2n_{\max} v_{\max} \sum_{j=1}^{N} \sqrt{b_{t-1}} \hat{\sigma}_j^c(x_{j,t})}{N c_{\min}(c_{\min} - \sqrt{b_T}\sigma)}, \tag{14}$$

where $\sum_{j=1}^{N} v_j \hat{n}_j(x_{j,t}) \geq \sum_{j=1}^{N} v_j n_j(x_j^*)$ by definition of the GCB selection rule, $v_{\max} := \max_{j=1}^{N} v_j$, and we assume that $c_{\min} - \sqrt{b_T}\sigma > 0$.

Using arguments similar to what has been used to bound the instantaneous regret r_t in Srinivas et al. [24] and Accabi et al. [1], and summing over the time horizon T, provides the final statement of the theorem. □

A.3 GCB$_{\mathsf{safe}}$ Analysis (Complete Proofs)

Theorem 6 (GCB$_{\mathsf{safe}}$ safety). *Given $\delta \in (0, 1)$, GCB$_{\mathsf{safe}}$ applied to the problem in Equations* (1a)–(1c) *is δ-safe and, therefore, the number of constraints violations is constant in T.*

Proof. Let us focus on a specific day t. Constraints (1b) and (1c) are satisfied by the solution of $\text{Opt}(\boldsymbol{\mu}, \lambda)$ for the properties of the optimization procedure. Define $\underline{n}_j(x_{j,t}) := \hat{n}_j(x_{j,t}) - \sqrt{b_{t-1}} \hat{\sigma}_j^n(x_{j,t})$. Thanks to the specific construction of the upper bounds we have that $c_j(x_{j,t}) \leq \bar{c}_j(x_{j,t})$ and $n_j(x_{j,t}) \geq \underline{n}_j(x_{j,t})$, each holding with probability at least $1 - \frac{3\delta}{\pi^2 NQTt^2}$. As a consequence, we have:

$$\frac{\sum_{j=1}^{N} v_j \; n_j(x_{j,t})}{\sum_{j=1}^{N} \; c_j(x_{j,t})} > \frac{\sum_{j=1}^{N} v_j \; \underline{n}_j(x_{j,t})}{\sum_{j=1}^{N} \; \bar{c}_j(x_{j,t})} \geq \lambda$$

and

$$\sum_{j=1}^{N} c_j(x_{j,t}) < \sum_{j=1}^{N} \bar{c}_j(x_{j,t}) \leq \beta.$$

Using a union bound over:

- the two GPs (number of clicks and costs);
- the time horizon T;
- the number of times each bid is chosen in a subcampaign (at most t);
- the number of arms present in each subcampaign ($|X_j|$);
- the number of subcampaigns (N);

we have:

$$\sum_{t=1}^{T} \mathbb{P}\left(\frac{\sum_{j=1}^{N} v_j \; n_j(\hat{x}_{j,t})}{\sum_{j=1}^{N} \; c_j(\hat{x}_{j,t})} < \lambda \vee \sum_{j=1}^{N} c_j(\hat{x}_{j,t}) > \beta \right) \leq 2 \sum_{j=1}^{N} \sum_{k=1}^{|X_j|} \sum_{h=1}^{T} \sum_{l=1}^{t} \frac{3\delta}{\pi^2 NQTl^2} \tag{15}$$

$$\leq 2 \sum_{j=1}^{N} \sum_{k=1}^{Q} \sum_{h=1}^{T} \sum_{l=1}^{+\infty} \frac{3\delta}{\pi^2 NQTl^2} = \delta. \tag{16}$$

$$\tag{17}$$

This concludes the proof. □

Theorem 7 (GCB$_{\mathsf{safe}}$ pseudo-regret). *Given $\delta \in (0, 1)$, GCB$_{\mathsf{safe}}$ applied to the problem in Equations* (1a)–(1c) *suffers from a pseudo-regret $R_t(GCB_{\mathsf{safe}}) = \Theta(T)$.*

Proof. The optimal solution has at least one of the constraints which is active, *i.e.*, it has the left-hand side equal to the right-hand side. Assume that the optimal clairvoyant solution $\left\{x_j^*\right\}_{j=1}^{N}$ to the optimization problem has a value of the ROI λ_{opt} equal to λ. We showed in the proof of Theorem 6 that for any allocation, with probability at least $1 - \frac{3\delta}{\pi^2 NQTt^2}$, it holds that $\frac{\sum_{j=1}^{N} v_j \; n_j(x_{j,t})}{\sum_{j=1}^{N} \; c_j(x_{j,t})} > \frac{\sum_{j=1}^{N} v_j \; \underline{n}_j(x_{j,t})}{\sum_{j=1}^{N} \; \bar{c}_j(x_{j,t})}$. This is true also for the optimal clairvoyant solution $\left\{x_j^*\right\}_{j=1}^{N}$, for which $\lambda = \frac{\sum_{j=1}^{N} v_j \; n_j(x^*)}{\sum_{j=1}^{N} \; c_j(x^*)} > \frac{\sum_{j=1}^{N} v_j \; \underline{n}_j(x^*)}{\sum_{j=1}^{N} \; \bar{c}_j(x^*)}$, implying that the values used in the ROI constraint make this allocation not feasible for the $\text{Opt}(\boldsymbol{\mu}, \lambda)$ procedure. As shown before, this happens with probability at least $1 - \frac{3\delta}{\pi^2 NQTt^2}$ at day t, and $1 - \delta$ over the time horizon T. To conclude, with probability $1 - \delta$, not depending on the time horizon T, we will not choose the optimal arm during the time horizon and, therefore, the regret of the algorithm cannot be sublinear. Notice that the same line of proof is also holding in the case the budget constraint is active, therefore, the previous result holds for each instance of the problem in Equations (1a)–(1c). □

THEOREM 8 ($\text{GCB}_{\mathsf{safe}}(\psi)$ PSEUDO-REGRET AND SAFETY WITH TOLERANCE). *When* $\psi \geq 2\frac{\beta_{opt}+n_{\max}}{\beta_{opt}^2}\sum_{j=1}^N v_j\sqrt{2\ln\left(\frac{\pi^2NQT^3}{3\delta'}\right)}\sigma$ *and* $\beta_{opt} < \beta\frac{\sum_{j=1}^N v_j}{\frac{N\,\beta_{opt}\psi}{\beta_{opt}+n_{\max}}+\sum_{j=1}^N v_j}$, *where* $\delta' \leq \delta$, β_{opt} *is the spend at the optimal solution of the original problem, and* $n_{\max} := \max_{j,x} n_j(x)$ *is the maximum over the sub-campaigns and the admissible bids of the expected number of clicks,* GCB_{safe} *provides a pseudo-regret w.r.t. the optimal solution to the original problem of* $O\left(\sqrt{T\sum_{j=1}^N \gamma_{j,T}}\right)$ *with probability at least* $1-\delta-\frac{\delta'}{QT^2}$, *while being* δ*-safe w.r.t. the constraints of the auxiliary problem.*

PROOF. In what follows, we show that, at a specific day t, since the optimal solution of the original problem $\left\{x_j^*\right\}_{j=1}^N$ is included in the set of feasible ones, we are in a setting analogous to the one of GCB, in which the regret is sublinear. Let us assume that the upper bounds on all the quantities (number of clicks and costs) holds. This has been shown before to occur with overall probability δ over the whole time horizon T. Moreover, notice that combining the properties of the budget of the optimal solution of the original problem β_{opt} and using $\psi = 2\frac{\beta_{opt}+n_{\max}}{\beta_{opt}^2}\sum_{j=1}^N v_j\sqrt{2\ln\left(\frac{\pi^2NQT^3}{3\delta'}\right)}\sigma$, we have:

$$\beta_{opt} < \beta\frac{\sum_{j=1}^N v_j}{\frac{N\,\beta_{opt}\psi}{\beta_{opt}+n_{\max}}+\sum_{j=1}^N v_j} \tag{18}$$

$$\left(\frac{N\,\beta_{opt}\psi}{\beta_{opt}+n_{\max}}+\sum_{j=1}^N v_j\right)\beta_{opt} < \beta\sum_{j=1}^N v_j \tag{19}$$

$$2N\sum_{j=1}^N v_j\sqrt{2\ln\left(\frac{\pi^2NQT^3}{3\delta'}\right)}\sigma + \sum_{j=1}^N v_j\beta_{opt} < \beta\sum_{j=1}^N v_j \tag{20}$$

$$\beta > \beta_{opt} + 2N\sqrt{2\ln\left(\frac{\pi^2NQT^3}{3\delta'}\right)}\sigma. \tag{21}$$

First, let us evaluate the probability that the optimal solution is not feasible. This occurs if its bounds are either violating the ROI or budget constraints. First, we show that analysing the budget constraint, the optimal solution of the original problem is feasible with high probability. Formally, it is not feasible with probability:

$$\mathbb{P}\left(\sum_{j=1}^N \overline{c}_j(x^*) > \beta\right) \leq \mathbb{P}\left(\sum_{j=1}^N \overline{c}_j(x^*) > \beta_{opt} + 2N\sqrt{2\ln\left(\frac{\pi^2NQT^3}{3\delta'}\right)}\sigma\right) \tag{22}$$

$$= \mathbb{P}\left(\sum_{j=1}^N \overline{c}_j(x^*) > \sum_{j=1}^N c_j(x^*) + 2N\sqrt{2\ln\frac{\pi^2NQT^3}{3\delta'}}\sigma\right) \tag{23}$$

$$\leq \sum_{j=1}^N \mathbb{P}\left(\overline{c}_j(x^*) > c_j(x^*) + 2\sqrt{2\ln\frac{\pi^2NQT^3}{3\delta'}}\sigma\right) \tag{24}$$

$$= \sum_{j=1}^N \mathbb{P}\left(\hat{c}_{j,t-1}(x^*) - c_j(x^*) > -\sqrt{b_t}\hat{\sigma}^c_{j,t-1}(x^*) + 2\sqrt{2\ln\frac{\pi^2NQT^3}{3\delta'}}\sigma\right) \tag{25}$$

$$\leq \sum_{j=1}^N \mathbb{P}\left(\hat{c}_{j,t-1}(x^*) - c_j(x^*) > \sqrt{2\ln\frac{\pi^2NQT^3}{3\delta'}}\hat{\sigma}^c_{j,t-1}(x^*)\right) \tag{26}$$

$$\leq \sum_{j=1}^N \mathbb{P}\left(\frac{\hat{c}_{j,t-1}(x^*) - c_j(x^*)}{\hat{\sigma}^c_{j,t-1}(x^*)} > \sqrt{2\ln\frac{\pi^2NQT^3}{3\delta'}}\right) \tag{27}$$

$$\leq \sum_{j=1}^N \frac{3\delta'}{\pi^2NQT^3} = \frac{3\delta'}{\pi^2QT^3}, \tag{28}$$

where, in the inequality in Equation (22) we used Equation (21), in Equation (27) we used the fact that $\frac{\pi^2NQt^2T}{3\delta} \leq \frac{\pi^2NQT^3}{3\delta'}$ for each $t \in \{1,\ldots,T\}$, $\hat{\sigma}^c_{j,t-1}(x^*) \leq \sigma$ for each j and t, and the inequality in Equation (28) is from Srinivas et al. [24]. Summing over the time horizon

T, we get that the optimal solution of the original problem $\left\{x_j^*\right\}_{j=1}^N$ is excluded from the set of the feasible ones with probability at most $\frac{3\delta'}{\pi^2 QT^2}$.

Second, we derive a bound over the probability that the optimal solution of the original problem is feasible due to the newly defined ROI constraint. Let us notice that since the ROI constraint is active we have $\lambda = \lambda_{opt}$. The probability that $\left\{x_j^*\right\}_{j=1}^N$ is not feasible due to the ROI constraint is:

$$\mathbb{P}\left(\frac{\sum_{j=1}^N v_j\ \underline{n}_j(x^*)}{\sum_{j=1}^N\ \overline{c}_j(x^*)} < \lambda - \psi\right) \tag{29}$$

$$\leq \mathbb{P}\left(\frac{\sum_{j=1}^N v_j\ \underline{n}_j(x^*)}{\sum_{j=1}^N\ \overline{c}_j(x^*)} < \lambda_{opt} - 2\frac{\beta_{opt}+n_{\max}}{\beta_{opt}^2}\sum_{j=1}^N v_j\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma\right) \tag{30}$$

$$= \mathbb{P}\left(\frac{\sum_{j=1}^N v_j\ \underline{n}_j(x^*)}{\sum_{j=1}^N\ \overline{c}_j(x^*)} < \frac{\sum_{j=1}^N v_j\ n_j(x^*)}{\sum_{j=1}^N\ c_j(x^*)} - 2\frac{\beta_{opt}+n_{\max}}{\beta_{opt}^2}\sum_{j=1}^N v_j\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma\right) \tag{31}$$

$$= \mathbb{P}\Bigg(\sum_{j=1}^N c_j(x^*)\sum_{j=1}^N v_j\ \underline{n}_j(x^*) < \sum_{j=1}^N\ \overline{c}_j(x^*)\sum_{j=1}^N v_j\ n_j(x^*)$$
$$-2\frac{\beta_{opt}+n_{\max}}{\beta_{opt}^2}\sum_{j=1}^N c_j(x^*)\sum_{j=1}^N\ \overline{c}_j(x^*)\sum_{j=1}^N v_j\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma\Bigg) \tag{32}$$

$$= \mathbb{P}\Bigg(\sum_{j=1}^N c_j(x^*)\sum_{j=1}^N v_j\ \underline{n}_j(x^*) - \sum_{j=1}^N c_j(x^*)\sum_{j=1}^N v_j\ n_j(x^*) +$$
$$\frac{2}{\beta_{opt}}\sum_{j=1}^N c_j(x^*)\sum_{j=1}^N\ \overline{c}_j(x^*)\sum_{j=1}^N v_j\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma$$
$$+\sum_{j=1}^N c_j(x^*)\sum_{j=1}^N v_j\ n_j(x^*) - \sum_{j=1}^N\ \overline{c}_j(x^*)\sum_{j=1}^N v_j\ n_j(x^*) +$$
$$\frac{2n_{\max}}{\beta_{opt}^2}\sum_{j=1}^N c_j(x^*)\sum_{j=1}^N\ \overline{c}_j(x^*)\sum_{j=1}^N v_j\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma < 0\Bigg) \tag{33}$$

$$\leq \mathbb{P}\left(\sum_{j=1}^N v_j\ \underline{n}_j(x^*) - \sum_{j=1}^N v_j\ n_j(x^*) + 2\underbrace{\frac{\sum_{j=1}^N\ \overline{c}_j(x^*)}{\beta_{opt}}}_{\geq 1}\sum_{j=1}^N v_j\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma < 0\right)$$
$$+\mathbb{P}\Bigg(\sum_{j=1}^N c_j(x^*)\sum_{j=1}^N v_j\ n_j(x^*) - \sum_{j=1}^N\ \overline{c}_j(x^*)\sum_{j=1}^N v_j\ n_j(x^*)$$
$$+2\underbrace{\frac{\sum_{j=1}^N c_j(x^*)\sum_{j=1}^N\ \overline{c}_j(x^*)}{\beta_{opt}^2}}_{\geq 1}\sum_{j=1}^N v_j\ \underbrace{n_{\max}}_{\geq n_j(x^*)}\ \sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma < 0\Bigg) \tag{34}$$

$$\leq \sum_{j=1}^N \mathbb{P}\left(\underline{n}_j(x^*) - n_j(x^*) + 2\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma \leq 0\right)$$
$$+\sum_{j=1}^N \mathbb{P}\left(c_j(x^*) - \overline{c}_j(x^*) + 2\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma < 0\right) \tag{35}$$

$$\leq \sum_{j=1}^{N} \mathbb{P}\left(\hat{n}_{j,t-1}(x^*) - \sqrt{b_t}\hat{\sigma}^n_{j,t-1}(x^*) - n_j(x^*) + 2\underbrace{\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma}_{\geq \sqrt{b_t}\hat{\sigma}^n_{j,t-1}(x^*)} < 0 \right)$$

$$+ \sum_{j=1}^{N} \mathbb{P}\left(c_j(x^*) - \hat{c}_{j,t-1}(x^*) - \sqrt{b_t}\hat{\sigma}^c_{j,t-1}(x^*) + 2\underbrace{\sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\sigma}_{\geq \sqrt{b_t}\hat{\sigma}^c_{j,t-1}(x^*)} < 0 \right) \tag{36}$$

$$\leq \sum_{j=1}^{N} \mathbb{P}\left(n_j(x^*) < \hat{n}_{j,t-1}(x^*) + \sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\hat{\sigma}^n_{j,t-1}(x^*) \right)$$

$$+ \sum_{j=1}^{N} \mathbb{P}\left(c_j(x^*) < \hat{c}_{j,t-1}(x^*) - \sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}}\hat{\sigma}^c_{j,t-1}(x^*) \right) \tag{37}$$

$$= \sum_{j=1}^{N} \mathbb{P}\left(\frac{n_j(x^*) - \hat{n}_{j,t-1}(x^*)}{\hat{\sigma}^n_{j,t-1}(x^*)} > \sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}} \right)$$

$$+ \sum_{j=1}^{N} \mathbb{P}\left(\frac{\hat{c}_{j,t-1}(x^*) - c_j(x^*)}{\hat{\sigma}^c_{j,t-1}(x^*)} > \sqrt{2\ln\frac{\pi^2 NQT^3}{3\delta'}} \right) \tag{38}$$

$$\leq 2\sum_{j=1}^{N} \frac{3\delta'}{\pi^2 NQT^3} = \frac{6\delta'}{\pi^2 QT^3}, \tag{39}$$

where in Equation (37) we used the fact that $\frac{\pi^2 NQt^2T}{3\delta} \leq \frac{\pi^2 NQT^3}{3\delta'}$ for each $t \in \{1, \dots, T\}$, $\hat{\sigma}^n_{j,t-1}(x^*) \leq \sigma$ for each j and t, and the inequality in Equation (39) is from Srinivas et al. [24]. Summing over the time horizon T ensures that the optimal solution of the original problem $\left\{x^*_j\right\}_{j=1}^N$ is excluded from the feasible solutions at most with probability $\frac{6\delta'}{\pi^2 QT^2}$. Finally, using a union bound, we have that the optimal solution can be chosen over the time horizon with probability at least $1 - \frac{3\delta'}{\pi^2 QT^2} - \frac{6\delta'}{\pi^2 QT^2} \leq 1 - \frac{\delta'}{QT^2}$.

Notice that here we want to compute the regret of the $\mathrm{GCB}_{\mathtt{safe}}$ algorithm w.r.t. $\left\{x^*_j\right\}_{j=1}^N$ which is not optimal for the analysed relaxed problem. Nonetheless, the proof on the pseudo-regret provided in Accabi et al. [1] is valid also for suboptimal solutions in the case it is feasible with high probability. This can be trivially shown using the fact that the regret w.r.t. a generic solution cannot be larger than the one computed w.r.t. the optimal one. Thanks to that, using a union bound over the probability that the bounds hold and that $\left\{x^*_j\right\}_{j=1}^N$ is feasible, we conclude that with probability at least $1 - \delta - \frac{\delta'}{QT^2}$ the regret $\mathrm{GCB}_{\mathtt{safe}}$ is of the order of $O\left(\sqrt{T\sum_{j=1}^N \gamma_{j,T}}\right)$. Finally, thanks to the property of the $\mathrm{GCB}_{\mathtt{safe}}$ algorithm shown in Theorem 6, the learning policy is δ-safe for the relaxed problem. □

In the case the active constraint is the one related to the budget we slightly relax it, substituting β with $\beta + \phi$.

Theorem 10 ($\mathrm{GCB}_{\mathtt{safe}}$ pseudo-regret and safety with tolerance). *When $\phi \geq 2N\sqrt{2\ln\left(\frac{\pi^2 NQT^3}{3\delta'}\right)}\sigma$, and $\lambda_{opt} > \lambda + \frac{(\beta + n_{\max})\phi \sum_{j=1}^N v_j}{N\beta^2}$, where $\delta' \leq \delta$, and $n_{\max} := \max_{j,x} n_j(x)$ is maximum expected number of clicks, $GCB_{\mathtt{safe}}$ provides a pseudo-regret w.r.t. the optimal solution to the original problem of $O\left(\sqrt{T\sum_{j=1}^N \gamma_{j,T}}\right)$ with probability at least $1 - \delta - \frac{6\delta'}{\pi^2 QT^2}$, while being δ-safe w.r.t. the constraints of the auxiliary problem.*

Proof. We show that at a specific day t since the optimal solution of the original problem $\left\{x^*_j\right\}_{j=1}^N$ is included in the set of feasible ones, we are in a setting analogous to the one of GCB, in which the regret is sublinear. Let us assume that the upper bounds on all the quantities (number of clicks and costs) holds. This has been shown before to occur with overall probability δ over the whole time horizon T.

First, let us evaluate the probability that the optimal solution is not feasible. This occurs if its bounds are either violating the ROI or budget constraints. From the fact that the ROI of the optimal solution satisfies $\lambda_{opt} > \lambda + \frac{(\beta+n_{\max})\phi \sum_{j=1}^{N} v_j}{N\beta^2}$, we have:

$$\mathbb{P}\left(\frac{\sum_{j=1}^{N} v_j \; \underline{n}_j(x^*)}{\sum_{j=1}^{N} \; \overline{c}_j(x^*)} < \lambda\right) \tag{40}$$

$$\leq \mathbb{P}\left(\frac{\sum_{j=1}^{N} v_j \; \underline{n}_j(x^*)}{\sum_{j=1}^{N} \; \overline{c}_j(x^*)} < \lambda_{opt} - \frac{(\beta + n_{\max})\phi \sum_{j=1}^{N} v_j}{N\beta^2}\right) \tag{41}$$

$$= \mathbb{P}\left(\frac{\sum_{j=1}^{N} v_j \; \underline{n}_j(x^*)}{\sum_{j=1}^{N} \; \overline{c}_j(x^*)} < \frac{\sum_{j=1}^{N} v_j \; n_j(x^*)}{\sum_{j=1}^{N} \; c_j(x^*)} - 2\frac{\beta_{opt} + n_{\max}}{\beta_{opt}^2} \sum_{j=1}^{N} v_j \sqrt{\ln \frac{\pi^2 NQT^3}{3\delta'}}\sigma\right) \tag{42}$$

$$\leq \frac{3\delta'}{\pi^2 QT^3}, \tag{43}$$

where the derivation used arguments similar to the ones applied in the proof for the ROI constraint in Theorem 8. Summing over the time horizon T ensures that the optimal solution of the original problem $\left\{x_j^*\right\}_{j=1}^{N}$ is excluded from the feasible solutions at most with probability $\frac{3\delta'}{\pi^2 QT^2}$.

Second, let us evaluate the probability for which the optimal solution of the original problem $\left\{x_j^*\right\}_{j=1}^{N}$ is excluded due to the budget constraint, formally:

$$\mathbb{P}\left(\sum_{j=1}^{N} \overline{c}_j(x^*) > \beta + \phi\right) \tag{44}$$

$$\leq \mathbb{P}\left(\sum_{j=1}^{N} \overline{c}_j(x^*) > \beta + 2N\sqrt{2\ln \frac{\pi^2 NQT^3}{3\delta'}}\sigma\right) \tag{45}$$

$$= \mathbb{P}\left(\sum_{j=1}^{N} \overline{c}_j(x^*) > \sum_{j=1}^{N} \; c_j(x^*) + 2N\sqrt{2\ln \frac{\pi^2 NQT^3}{3\delta'}}\sigma\right) \tag{46}$$

$$\leq \sum_{j=1}^{N} \mathbb{P}\left(\overline{c}_j(x^*) > c_j(x^*) + 2\sqrt{\ln \frac{12NT^3}{\pi^2\delta'}}\sigma\right) \tag{47}$$

$$= \sum_{j=1}^{N} \mathbb{P}\left(\hat{c}_{j,t-1}(x^*) - c_j(x^*) \geq -\sqrt{b_t}\hat{\sigma}^c_{j,t-1}(x^*) + 2\sqrt{2\ln \frac{\pi^2 NQT^3}{3\delta'}}\sigma\right) \tag{48}$$

$$\leq \sum_{j=1}^{N} \mathbb{P}\left(\hat{c}_{j,t-1}(x^*) - c_j(x^*) \geq \sqrt{2\ln \frac{\pi^2 NQT^3}{3\delta'}}\hat{\sigma}^c_{j,t-1}(x^*)\right) \tag{49}$$

$$\leq \sum_{j=1}^{N} \mathbb{P}\left(\frac{\hat{c}_{j,t-1}(x^*) - c_j(x^*)}{\hat{\sigma}^c_{j,t-1}(x^*)} \geq \sqrt{2\ln \frac{\pi^2 NQT^3}{3\delta'}}\right) \tag{50}$$

$$\leq \sum_{j=1}^{N} \frac{3\delta'}{\pi^2 NQT^3} = \frac{3\delta'}{\pi^2 QT^3}, \tag{51}$$

where we use the fact that $\beta = \beta_{opt}$, and the derivation used arguments similar to the ones applied in the proof for the budget constraint in Theorem 8. Summing over the time horizon T, we get that the optimal solution of the original problem $\left\{x_j^*\right\}_{j=1}^{N}$ is excluded from the set of the feasible ones with probability at most $\frac{\pi^2\delta'}{6T^2}$. Finally, using a union bound, we have that the optimal solution can be chosen over the time horizon with probability at least $1 - \frac{3\delta'}{\pi^2 QT^2}$.

Notice that here we want to compute the regret of the $\text{GCB}_{\mathtt{safe}}$ algorithm w.r.t. $\left\{x_j^*\right\}_{j=1}^{N}$ which is not optimal for the analysed relaxed problem. Nonetheless, the proof on the pseudo-regret provided in Accabi et al. [1] is valid also for suboptimal solutions in the case it is feasible with high probability. This can be trivially shown using the fact that the regret w.r.t. a generic solution cannot be larger than the one computed on the optimal one. Thanks to that, using a union bound over the probability that the bounds hold and that $\left\{x_j^*\right\}_{j=1}^{N}$ is feasible, we conclude that with probability at least $1-\delta-\frac{6\delta'}{\pi^2 QT^2}$ the regret GCB$_{\texttt{safe}}$ is of the order of $O\left(\sqrt{T\sum_{j=1}^{N}\gamma_{j,T}}\right)$. Finally, thanks to the property of the GCB$_{\texttt{safe}}$ algorithm shown in Theorem 6, the learning policy is δ-safe for the relaxed problem. □

A final case occurs when both the constraints are active. In this setting the relaxation should be performed on both constraints, i.e., we need to set the value of λ to $\lambda+\psi$ and the value β to $\beta+\phi$ in the original optimization problem.[11]

Theorem 11 (GCB$_{\texttt{safe}}$ pseudo-regret for the ROI and budget relaxed problem). *Setting* $\psi = 2\frac{\beta_{opt}+n_{\max}}{\beta_{opt}^2}\sum_{j=1}^{N} v_j \sqrt{2\ln\left(\frac{\pi^2 NQT^3}{3\delta'}\right)}\sigma$ *and* $\phi = 2N\sqrt{2\ln\left(\frac{\pi^2 NQT^3}{3\delta'}\right)}\sigma$, *where* $\delta' \leq \delta$, *GCB*$_{\texttt{safe}}$ *provides a pseudo-regret w.r.t. the optimal solution to the original problem of* $O\left(\sqrt{T\sum_{j=1}^{N}\gamma_{j,T}}\right)$ *with probability at least* $1-\delta-\frac{\delta'}{QT^2}$, *while being* δ*-safe w.r.t. the constraints of the auxiliary problem.*

Proof. The proof follows from combining the arguments about the ROI constraint used in Theorem 8 and those about the budget constraint used in Theorem 10. □

[11]Notice that this approach might be applied also in the case we are not aware of which constraint is active or if the optimal solution does not satisfy the requirements stated in Theorem 8 and 10.

B ADDITIONAL EXPERIMENTS FOR THE PAPER "SAFE ONLINE BID OPTIMIZATION WITH UNCERTAIN RETURN-ON-INVESTMENT AND BUDGET CONSTRAINTS"

In this section we provide additional information to allow full reproducibility of the experiments provided in the main paper.

B.1 Parameters and Setting of Experiment #1

The code has been run on a Intel(R) Core(TM) $i7-4710MQ$ CPU with 16 GiB of system memory. The operating system was Ubuntu 18.04.5 LTS, and the experiments have been run on Python 3.7.6. The libraries used in the experiments, with the corresponding version were:

- `matplotlib==3.1.3`
- `gpflow==2.0.5`
- `tikzplotlib==0.9.4`
- `tf_nightly==2.2.0.dev20200308`
- `numpy==1.18.1`
- `tensorflow_probability==0.10.0`
- `scikit_learn==0.23.2`
- `tensorflow==2.3.0`

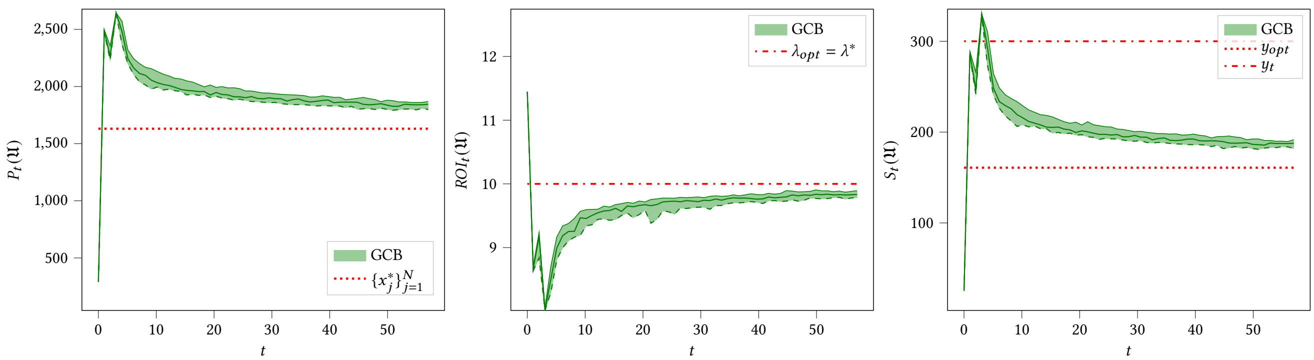
On this architecture, the average execution time of the each algorithm takes an average of $\approx$ 30 sec for each day t of execution. Table 1 specifies the values of the parameters of cost and number-of-click functions of the subcampaigns used in Experiment #1.

Table 1: Parameters of the synthetic settings used in Experiment #1.

	C_1	C_2	C_3	C_4	C_5
θ_j	60	77	75	65	70
δ_j	0.41	0.48	0.43	0.47	0.40
α_j	497	565	573	503	536
γ_j	0.65	0.62	0.67	0.68	0.69
σ_f GP revenue	0.669	0.499	0.761	0.619	0.582
l GP revenue	0.425	0.469	0.471	0.483	0.386
σ_f GP cost	0.311	0.443	0.316	0.349	0.418
l GP cost	0.76	0.719	0.562	0.722	0.727

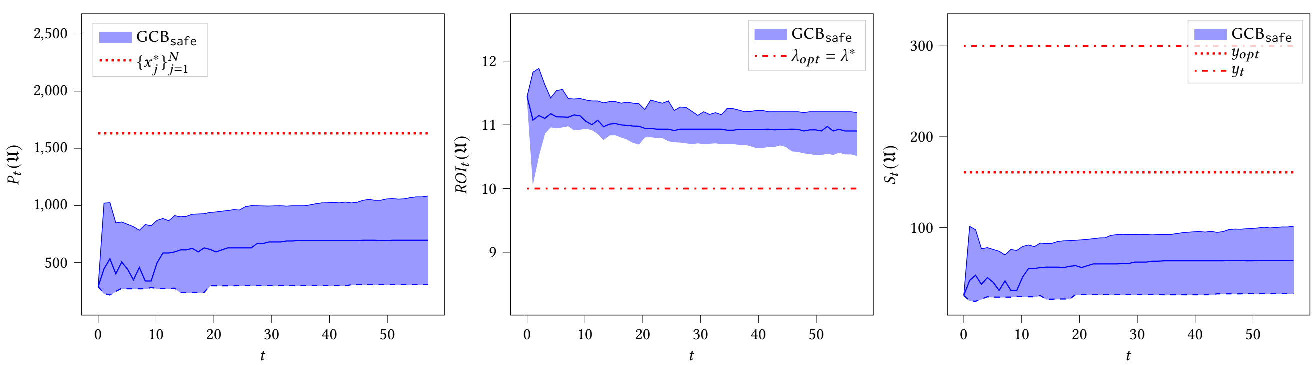
B.2 Additional Figures Experiment #2

In Figures **??**, **??**, and **??** we report the 90% and 10% of the quantities analysed in the experimental section for Experiment #2 provided by the GCB, $\text{GCB}_{\texttt{safe}}$, and $\text{GCB}_{\texttt{safe}}(0.05)$, respectively. These results show that the constraints are satisfied by $\text{GCB}_{\texttt{safe}}$, and $\text{GCB}_{\texttt{safe}}(0.05)$ also with high probability. While for $\text{GCB}_{\texttt{safe}}$ this is expected due to the theoretical results we provided, the fact that also $\text{GCB}_{\texttt{safe}}(0.05)$ guarantees safety w.r.t. the original optimization problem suggests that in some specific setting $\text{GCB}_{\texttt{safe}}$ is too conservative. This is reflected in a lower cumulative revenue, which might be negative from a business point of view.



(a) (b) (c)

Figure 4: Results of Experiment #3: daily revenue (a), ROI (b), and spend (c) obtained by GCB in Experiment 3. Dash-dotted lines correspond to the optimum values for the revenue and ROI, while dashed lines correspond to the values of the ROI and budget constraints.

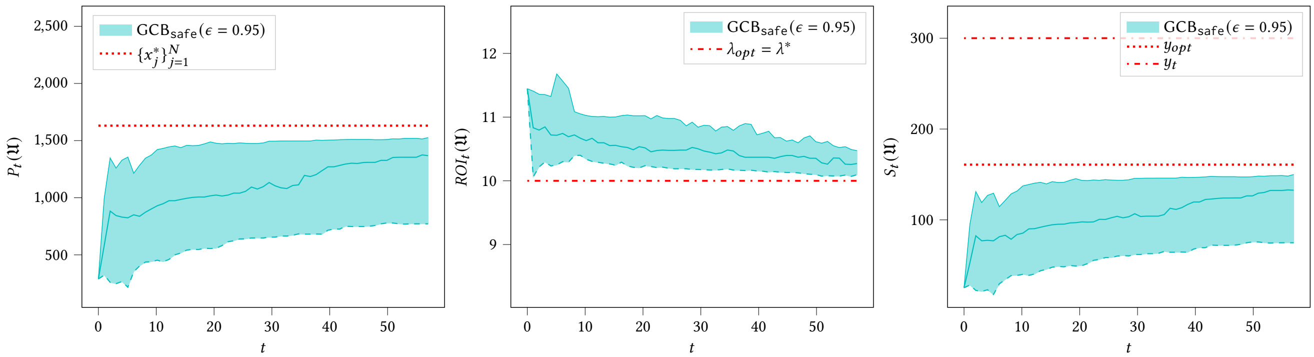


(a) (b) (c)

Figure 5: Results of Experiment #3: daily revenue (a), ROI (b), and spend (c) obtained by $\text{GCB}_{\texttt{safe}}$ in Experiment 3. Dash-dotted lines correspond to the optimum values for the revenue and ROI, while dashed lines correspond to the values of the ROI and budget constraints.

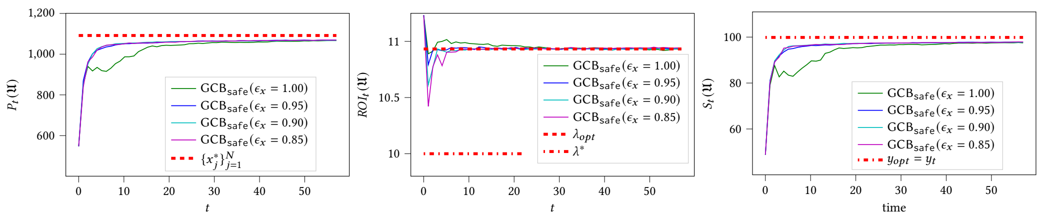
B.3 Experiment #3

In real-world scenarios, the business goals in terms of volumes-profitability tradeoff are often blurred, and sometimes can be desirable to slightly violate the constraints (usually, the ROI constraint) in favor of a significant volumes increase. However, analyzing and acquiring information about these tradeoff curves requires to explore volumes opportunities by relaxing the constraints. In this experiment, we show how our approach can be adjusted to address this problem in practice. We use the same setting of Experiment #1, except for the input we pass to the $\text{GCB}_{\texttt{safe}}$ algorithm. More precisely, we relax the ROI constraint by a value $\psi \in \{0, 0.05, 0.1, 0.15\}$, and we run 4 instances of $\text{GCB}_{\texttt{safe}}$ each associated to a different ψ value. Notice that $\text{GCB}_{\texttt{safe}}(0)$ corresponds to the use of $\text{GCB}_{\texttt{safe}}$ in the original problem, *i.e.*, consists in the application of $\text{GCB}_{\texttt{safe}}$ without any relaxation of the ROI constraint. As a result, except for the first instance, we allow



(a) (b) (c)

Figure 6: Results of Experiment #3: daily revenue (a), ROI (b), and spend (c) obtained by GCB($\epsilon_x = 0.95$). Dash-dotted lines correspond to the optimum values for the revenue and ROI, while dashed lines correspond to the values of the ROI and budget constraints.

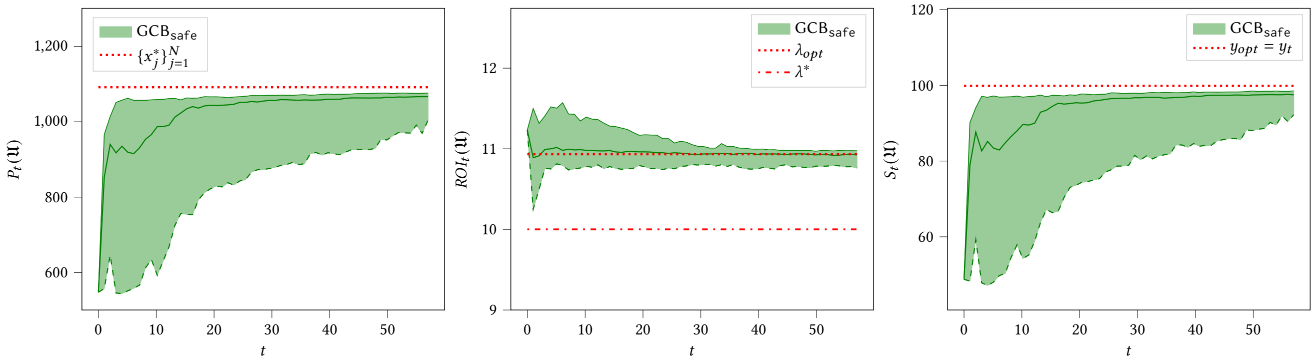


(a) (b) (c)

Figure 7: Results of Experiment #3: Median values of the daily revenue (a), ROI (b) and spend (c) obtained by GCB$_{\mathsf{safe}}$ with different values of ψ.

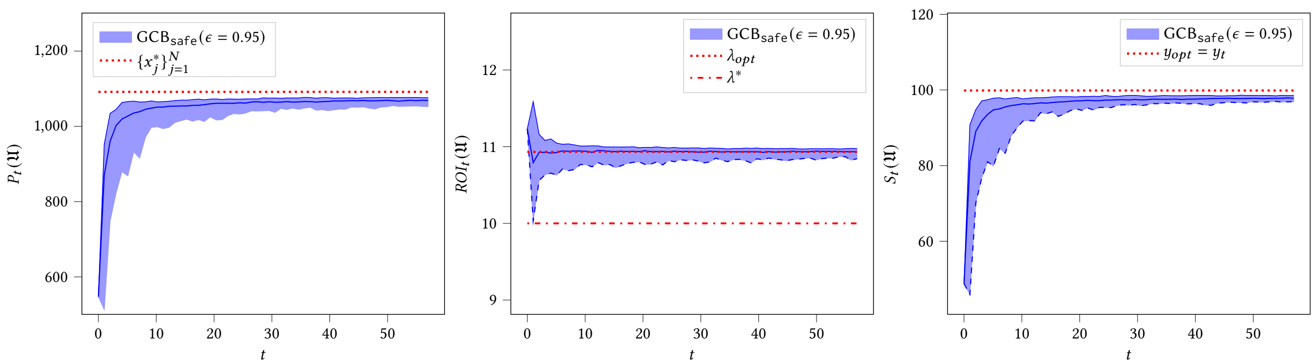
GCB$_{\mathsf{safe}}$ to violate the ROI constraint, but, with high probability, the violation is bounded by at most 5%, 10%, 15% of λ, respectively. Instead, we do not introduce any tolerance for the daily budget constraint β.

In Figure 7, we show the median values, on 100 independent runs, of the performance in terms of daily revenue, ROI, and spend of GCB$_{\mathsf{safe}}$ for every value of ψ. The 10% and 90% quantiles of these quantities are reported in Figure 8, 9 and 10. The results show that, in practice, allowing a small tolerance in the ROI constraint violation, we can improve the exploration and, therefore, lead to faster convergence. We note that if we set a value of $\psi \geq 0.05$, we achieve significantly better performance in the first learning steps ($t < 20$) still maintaining a robust behavior in terms of constraints violation. Most importantly, a small tolerance leads only to a violation of the ROI constraint in the early learning stages, but the behavior at convergence is the same obtained without any tolerance.



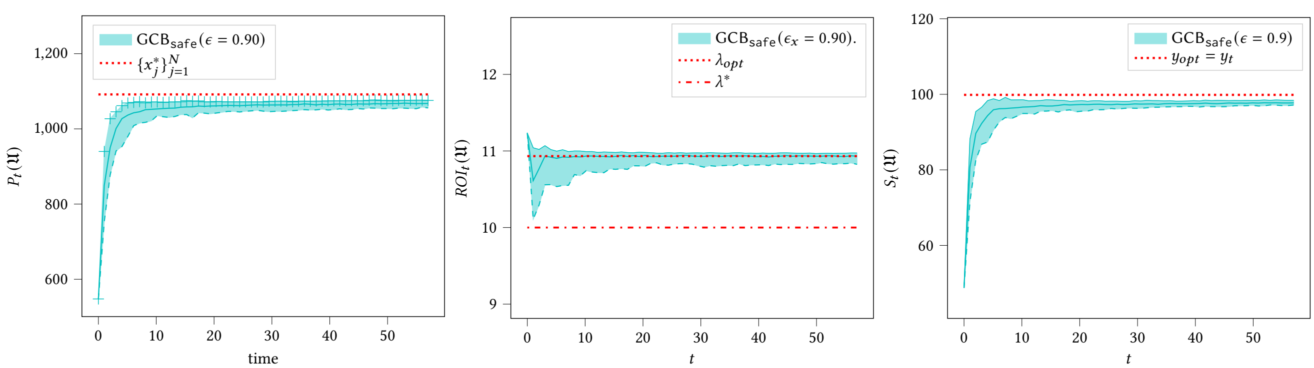
(a) (b) (c)

Figure 8: Results of Experiment #2: daily revenue (a), ROI (b), and spend (c) obtained by GCB$_{\text{safe}}$. Dash-dotted lines correspond to the optimum values for the revenue and ROI, while dashed lines correspond to the values of the ROI and budget constraints.



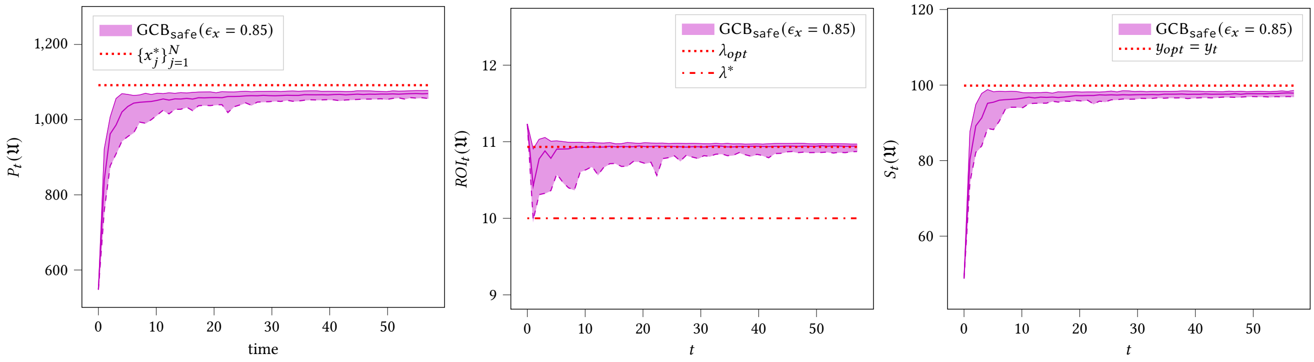
(a) (b) (c)

Figure 9: Results of Experiment #2: daily revenue (a), ROI (b), and spend (c) obtained by and GCB$_{\text{safe}}(\epsilon_x = 0.95)$. Dash-dotted lines correspond to the optimum values for the revenue and ROI, while dashed lines correspond to the values of the ROI and budget constraints.



(a) (b) (c)

Figure 10: Results of Experiment #2: daily revenue (a), ROI (b), and spend (c) obtained by and GCB$_{\text{safe}}(\epsilon_x = 0.90)$. Dash-dotted lines correspond to the optimum values for the revenue and ROI, while dashed lines correspond to the values of the ROI and budget constraints.



(a) **(b)** **(c)**

Figure 11: Results of Experiment #2: daily revenue (a), ROI (b), and spend (c) obtained by and GCB$_{\mathsf{safe}}(\epsilon_x = 0.85)$. Dash-dotted lines correspond to the optimum values for the revenue and ROI, while dashed lines correspond to the values of the ROI and budget constraints.

B.4 Experiment #4

In this experiment we extend the results of Experiment #1 and Experiment #3 to other settings. We simulate $N = 5$ subcampaigns with a daily budget $\beta = 100$, with $|X_j| = 201$ bid values evenly spaced in $[0,\ 2]$, $|Y| = 101$ cost values evenly spaced in $[0,\ 100]$, being the daily budget $\beta = 100$, and $|R|$ evenly spaced revenue values depending on the setting.

We build 10 scenarios that differ in the parameters defining the cost and revenue functions, and in the ROI parameter λ. Recall that the number-of-click functions coincides with the revenue functions since $v_j = 1$ for each $j \in \{1, \ldots, N\}$. Parameters $\alpha_j \in \mathbb{N}^+$ and $\theta_j \in \mathbb{N}^+$ are sampled from discrete uniform distributions $\mathcal{U}\{50, 100\}$ and $\mathcal{U}\{400, 700\}$, respectively. Parameters γ_j and δ_j are sampled from the continuous uniform distributions $\mathcal{U}[0.2, 1.1)$. Finally, parameters λ are chosen so that the ROI constraint would be an active constraint for the original problem. Table 2 summarize the values of such parameters.

Results. Table 3 reports the performances of algorithms GCB, $\texttt{GCB}_\texttt{safe}$, $\texttt{GCB}_\texttt{safe}(0.05)$ and $\texttt{GCB}_\texttt{safe}(0.10)$. In particular, $\mathbb{E}[\mathrm{CR}_{t=\hat{t}}]$ is the cumulative revenue until day $\hat{t}$ averaged on the number of simulations, while $\sigma_{\mathrm{CR}_{t=\hat{t}}}$ and $i_{th}^{t=\hat{t}}$p. are the corresponding standard deviation and i_{th} percentile, respectively. These results are reported w.r.t. two different time instant: $t = \lfloor \frac{T}{2} \rfloor = 28$, *i.e.,* at half of the period, and $t = T = 57$, *i.e.,* at the end of the time horizon. Finally, S_{ROI} and S_{budget} denotes the total number of days in which the ROI and the budget constraints were violated, respectively. In the last two columns we report the percentage of days on which the ROI and the budget constraint were violated, *i.e.,* $\frac{S_{ROI}}{T}$ and $\frac{S_{budget}}{T}$, respectively, averaged by the number of simulations. We performed 100 independent runs for each setting and each algorithm.

The results are in line with what have been observed in the main paper, showing that the $\texttt{GCB}_\texttt{safe}$ algorithm and its relaxed variants are able not to violate the constraints with high probability, while GCB shows the worst performance in terms of constraints violations. In terms of cumulative revenue, the algorithms providing the largest values are the ones violating the constraint, while the algorithm showing the largest revenue while satisfying the problem constraints is $\texttt{GCB}_\texttt{safe}$ with $\psi = 0.05$. These results corroborates the idea that the relaxing the constraints for a small percentage (*e.g.,* 5%) provides a good tradeoff between revenue maximization and constraint satisfaction in most of the cases.

Table 2: Parameters characterizing the 10 different settings in Experiment #4.

		C_1	C_2	C_3	C_4	C_5	λ
Setting 1	θ_j	530	417	548	571	550	10.0
	δ_j	0.356	0.689	0.299	0.570	0.245	
	α_j	83	97	72	100	96	
	γ_j	0.939	0.856	0.484	0.661	0.246	
Setting 2	θ_j	597	682	698	456	444	14.0
	δ_j	0.202	0.520	0.367	0.393	0.689	
	α_j	83	98	56	60	51	
	γ_j	0.224	0.849	0.726	0.559	0.783	
Setting 3	θ_j	570	514	426	469	548	10.5
	δ_j	0.217	0.638	0.694	0.391	0.345	
	α_j	97	78	53	80	82	
	γ_j	0.225	0.680	1.051	0.412	0.918	
Setting 4	θ_j	487	494	467	684	494	12.0
	δ_j	0.348	0.424	0.326	0.722	0.265	
	α_j	62	79	76	69	99	
	γ_j	0.460	1.021	0.515	0.894	1.056	
Setting 5	θ_j	525	643	455	440	600	14.0
	δ_j	0.258	0.607	0.390	0.740	0.388	
	α_j	52	87	68	99	94	
	γ_j	0.723	0.834	1.054	1.071	0.943	
Setting 6	θ_j	617	518	547	567	576	11.0
	δ_j	0.844	0.677	0.866	0.252	0.247	
	α_j	71	53	87	98	59	
	γ_j	0.875	0.841	1.070	0.631	0.288	
Setting 7	θ_j	409	592	628	613	513	11.5
	δ_j	0.507	0.230	0.571	0.359	0.307	
	α_j	77	78	91	50	71	
	γ_j	0.810	0.246	0.774	0.516	0.379	
Setting 8	θ_j	602	605	618	505	588	13.0
	δ_j	0.326	0.265	0.201	0.219	0.291	
	α_j	67	80	99	77	99	
	γ_j	0.671	0.775	0.440	0.310	0.405	
Setting 9	θ_j	486	684	547	419	453	13.0
	δ_j	0.418	0.330	0.529	0.729	0.679	
	α_j	53	82	58	96	100	
	γ_j	0.618	0.863	0.669	0.866	0.831	
Setting 10	θ_j	617	520	422	559	457	14.0
	δ_j	0.205	0.539	0.217	0.490	0.224	
	α_j	51	86	93	61	84	
	γ_j	1.0493	0.779	0.233	0.578	0.562	

Table 3: Performances of the GCB, GCB_{safe}, $GCB_{safe}(0.05)$, and $GCB_{safe}(0.10)$ algorithms in the 10 different settings in Experiment #4.

		$\mathbb{E}\left[CR_{t=\lfloor \frac{T}{2} \rfloor}\right]$	$\mathbb{E}[CR_{t=T}]$	$\sigma_{CR_{t=T}}$	$\sigma_{CR_{t=\lfloor \frac{T}{2} \rfloor}}$	$50_{th}^{t=T}$ p.	$50_{th}^{t=\lfloor \frac{T}{2} \rfloor}$ p.	$90_{th}^{t=T}$ p.	$90_{th}^{t=\lfloor \frac{T}{2} \rfloor}$ p.	$10_{th}^{t=T}$ p.	$10_{th}^{t=\lfloor \frac{T}{2} \rfloor}$ p.	$\mathbb{E}[\frac{S_{ROI}}{T}]$	$\mathbb{E}[\frac{S_{budget}}{T}]$
Setting 1	GCB	30767.336	57481.762	556.485	376.091	57497.185	30811.6042	58081.570	31239.227	56758.890	30288.910	1.000	0.617
	GCB_{safe}	21549.919	44419.090	4766.152	2474.262	45348.994	21972.152	46783.628	23163.449	42287.807	20324.750	0.019	0.000
	$GCB_{safe}(0.05)$	23524.149	48028.035	4902.964	2487.586	48626.046	23831.680	50388.713	24827.723	46307.675	22506.989	0.208	0.000
	$GCB_{safe}(0.10)$	25859.354	52327.738	829.526	611.281	52338.567	25887.959	53324.250	26605.853	51316.486	25104.946	0.938	0.002
Setting 2	GCB	35566.326	63664.204	1049.276	679.520	63701.943	35573.450	64984.421	36524.615	62249.768	34675.640	1.000	0.136
	GCB_{safe}	16290.759	34675.746	8541.501	4448.184	37028.783	17647.169	39594.699	19473.840	27748.425	11141.368	0.030	0.000
	$GCB_{safe}(0.05)$	19564.274	40962.919	6013.044	3122.532	41823.542	20152.608	44468.575	21698.836	38640.207	17645.532	0.042	0.000
	$GCB_{safe}(0.10)$	22099.617	46694.548	6382.710	3112.150	47749.883	22433.984	51564.023	24776.729	44099.097	19929.975	0.715	0.000
Setting 3	GCB	30213.282	54845.400	757.500	478.611	54816.940	30177.713	55734.991	30885.009	54006.505	29638.342	1.000	0.246
	GCB_{safe}	16577.752	35726.325	8239.174	4361.902	38302.025	18114.824	40746.921	19882.713	27279.764	8791.751	0.030	0.000
	$GCB_{safe}(0.05)$	18370.540	38757.228	8492.878	4594.693	41422.689	19808.168	43337.069	21092.243	30413.268	12678.440	0.067	0.000
	$GCB_{safe}(0.10)$	19993.574	42184.689	9652.822	5056.867	44820.418	21574.935	47659.265	23118.783	36570.721	14450.134	0.747	0.000
Setting 4	GCB	37383.516	71404.431	351.167	262.972	71399.582	37387.776	71877.052	37732.239	70930.123	37021.387	0.982	0.982
	GCB_{safe}	13817.172	29101.003	7052.947	3646.223	30992.413	14680.303	35602.210	17256.390	20509.269	9562.233	0.002	0.000
	$GCB_{safe}(0.05)$	18270.458	39802.784	10232.989	4955.693	38296.818	17994.578	53375.436	24962.574	25197.830	11341.536	0.008	0.000
	$GCB_{safe}(0.10)$	24095.853	51515.405	11094.352	5639.939	56621.600	24902.495	61992.866	30020.554	35642.967	16198.675	0.564	0.000
Setting 5	GCB	39523.308	74638.549	642.852	392.228	74693.882	39529.172	75405.263	40049.761	73756.512	39063.081	0.982	0.313
	GCB_{safe}	23230.524	48956.750	6715.177	3486.395	50021.031	23838.642	53644.831	26266.692	42946.297	19287.622	0.000	0.000
	$GCB_{safe}(0.05)$	27003.316	56205.696	2578.603	1742.997	56554.329	27211.923	58839.509	28802.539	53278.885	24987.319	0.000	0.000
	$GCB_{safe}(0.10)$	30207.289	63411.706	5636.641	2916.673	64364.864	30665.923	66764.410	32212.999	60519.219	28260.128	0.592	0.000
Setting 6	GCB	35775.780	67118.895	327.601	260.730	67130.819	35795.169	67536.732	36111.540	66726.744	35424.307	0.982	0.982
	GCB_{safe}	7707.891	14448.084	6006.165	3065.0127	15019.737	8075.216	18581.109	9800.874	6781.934	3926.408	0.022	0.000
	$GCB_{safe}(0.05)$	7710.433	14968.966	6174.049	2974.836	15161.566	8157.195	20548.672	10351.782	7954.149	3860.675	0.024	0.000
	$GCB_{safe}(0.10)$	15507.708	34716.555	16133.168	7280.176	37409.763	16601.218	55236.860	25366.308	9895.308	5188.600	0.189	0.000
Setting 7	GCB	35330.492	63038.578	873.627	401.328	63088.710	35367.147	64226.964	35793.889	61754.491	34823.374	1.000	0.408
	GCB_{safe}	14806.541	31662.450	5651.204	3090.871	33009.540	15570.468	35004.135	16922.489	28296.715	11338.293	0.037	0.000
	$GCB_{safe}(0.05)$	17606.829	37744.715	4173.089	2619.873	38321.168	18161.518	41184.415	19805.806	33914.967	15276.547	0.031	0.000
	$GCB_{safe}(0.10)$	20046.597	42528.584	7497.485	3624.829	43765.141	20683.409	47187.695	22301.568	38988.321	18314.219	0.696	0.000
Setting 8	GCB	42322.124	79571.510	476.880	375.810	79581.190	42317.922	80073.259	42743.107	78969.521	41913.760	1.000	0.982
	GCB_{safe}	22478.215	48046.987	11779.079	6000.067	52094.653	24180.975	57321.725	28024.487	30655.291	13338.036	0.020	0.000
	$GCB_{safe}(0.05)$	27477.419	58450.174	10296.288	5605.913	61404.554	28845.740	66902.032	32883.702	41196.805	18222.303	0.021	0.000
	$GCB_{safe}(0.10)$	33255.310	68252.406	3436.395	2417.770	68886.202	33857.704	70758.511	35377.143	65394.939	30696.023	0.069	0.000
Setting 9	GCB	37363.123	70280.744	672.557	347.550	70275.810	37352.860	71123.791	37811.648	69379.646	36942.376	1.000	0.339
	GCB_{safe}	18895.696	40116.370	5522.553	3047.034	40673.320	19357.076	43850.251	21161.005	37310.507	17222.155	0.028	0.000
	$GCB_{safe}(0.05)$	23683.390	51138.961	3110.432	2036.069	50984.031	23375.652	54545.527	26174.860	47465.626	21385.349	0.031	0.000
	$GCB_{safe}(0.10)$	29675.845	63574.004	3810.347	3323.916	64011.449	30112.566	66658.298	32559.645	60970.369	27280.237	0.795	0.000
Setting 10	GCB	41973.160	80570.847	435.533	344.547	80568.068	42019.380	81127.535	42388.800	80023.986	41496.669	1.000	0.982
	GCB_{safe}	28785.949	58965.259	3097.576	1465.835	60033.228	28917.795	62353.466	30535.428	54590.515	26931.753	0.018	0.000
	$GCB_{safe}(0.05)$	31004.511	63685.460	3787.176	1876.417	65273.877	31550.496	67364.597	33105.511	57860.165	28349.753	0.019	0.000
	$GCB_{safe}(0.10)$	33358.275	68480.403	4224.024	2181.878	70388.508	33998.870	72730.750	35838.227	61971.163	30317.769	0.652	0.000