# Online Optimization with Uncertain Information, an empirical study

Course project for CMPUT 676, University of Alberta

Roham Bahri

December, 2024

## 1 Introduction

In many real-world scenarios, decisions must be made sequentially as new information becomes available over time. This leads to the field of *online optimization*, where algorithms operate without full knowledge of future inputs. Such settings present a significant challenge: how to make robust decisions in the face of uncertainty while striving for optimal outcomes. Online optimization has been widely studied, with competitive analysis serving as a method for evaluating algorithm performance. In this framework, the performance of an online algorithm is measured against an optimal offline solution that has full knowledge of the input sequence. The worst-case ratio between the two outcomes, termed the *competitive ratio*, quantifies the algorithm's effectiveness in handling uncertainty.

However, competitive analysis often assumes an adversarial input, which can result in overly conservative algorithms. This limitation is particularly evident in applications where inputs exhibit unpredictable behavior or abrupt shifts. For example, in query allocation problem faced by search engines, the frequency of user queries can deviate drastically from expected patterns due to unforeseen events. Algorithms designed with rigid assumptions about input behavior may fail to adapt to such changes, leading to suboptimal performance.

To address these challenges, researchers have proposed hybrid approaches that balance optimistic and pessimistic strategies. Optimistic algorithms leverage prior knowledge or estimates about input patterns, achieving superior performance when these assumptions hold. In contrast, pessimistic algorithms focus on worst-case scenarios, ensuring robust performance even under adversarial conditions. By combining these perspectives, it is possible to design algorithms that dynamically adapt to varying input conditions, achieving a trade-off between adaptability and robustness.

In this project, we focus on the online query allocation problem, a generalization of the online matching problem, motivated by applications in online advertising. Search engines must dynamically allocate ad spaces to advertisers based on bids and budgets, often under uncertain query frequencies. While prior algorithms achieve strong competitive ratios without relying on input estimates, incorporating such estimates can unlock additional value. We build upon the framework introduced by Mahdian et al. (2012), which integrates both optimistic and pessimistic strategies, allowing algorithms to adapt to real-world input variability while maintaining competitive guarantees.

This report is structured as follows: Section 2 provides a detailed explanation of the problem setting and introduces the key concepts and notations used throughout the study. Section 3 describes the proposed algorithm, including its underlying assumptions, design, and implementation details. Section 4 outlines the experimental setup, including the datasets, evaluation metrics, and baseline algorithms used for comparison. Section 5 presents the results, offering an analysis of the algorithm's performance under various scenarios and parameter settings. Section 6 discusses the findings, highlighting the strengths, limitations, and implications of the proposed approach.

# 2 Problem Definition

The query allocation problem, first introduced by Mehta et al. (2007), generalizes the classic online matching problem to incorporate budget constraints. This model is particularly relevant for online advertising, where search engines allocate ad spaces to advertisers based on their bids and budgets.

In this problem, let $A$ be the set of advertisers and $K$ be the set of keywords. The algorithm receives a sequence of keyword queries from $K$ in an online fashion. Each advertiser $i \in A$ specifies a total budget $B_i$ and a bid $b_{ij}$ for each keyword $j \in K$. When a query for a keyword $j$ arrives, the algorithm assigns it to an advertiser $i$, charging the advertiser an amount equal to their bid $b_{ij}$.

The goal is to maximize the total revenue generated, defined as the sum of $b_{i(q),j(q)}$ over all queries $q$, where $i(q)$ is the advertiser assigned to query $q$, and $j(q)$ is the corresponding keyword. This maximization is subject to charging each advertiser at most its budget.

If $n_j$, the number of times keyword $j$ is queried, were known in advance, the problem could be formulated as the following maximization program:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i \in A, j \in K} b_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{i \in A} x_{ij} \leq n_j \quad \forall j \in K, \\
& \sum_{j \in K} b_{ij} x_{ij} \leq B_i \quad \forall i \in A, \\
& x_{ij} \geq 0 \quad \forall i \in A, \forall j \in K,
\end{aligned}
$$

where $x_{ij}$ represents the allocation of queries for keyword $j$ to advertiser $i$.

As noted by Mehta et al. (2007), the assumption that bids $b_{ij}$ are small relative to budgets $B_i$ is justified in practice. This simplifies the problem by allowing the relaxation of the integrality constraints on $x_{ij}$ without significantly affecting the solution.

# 3 Algorithm

This section presents the algorithmic framework for query allocation, building on the algorithm introduced by Mehta et al. (2007) with competitive ratio of $(1-1/e)$ and incorporating a parameterized extension to improve performance flexibility.

The original algorithm proposed by Mehta et al. (2007) dynamically adjusts the bids of advertisers based on their budget utilization. Specifically, it introduces a *discount factor* $\psi(x) = 1 - e^{-(1-x)}$. As queries arrive, the algorithm assigns each query to the advertiser $i \in A$ who maximizes the product of their bid $b_{ij}$ and $\psi(T(i))$, where $T(i) = m_i/B_i$ is the fraction of advertiser $i$'s budget $B_i$ that has been spent, and $m_i$ is the total amount spent by $i$ so far.

Mahdian et al. (2012) introduce a parameterized algorithm $Q(\alpha)$, which incorporates a tunable parameter $\alpha \geq 1$ to balance reliance on external recommendations and bid adjustments. They assume the algorithm has access to an algorithm $O$ that recommends an advertiser for each query. $O$ can be based on an algorithm that solves the linear program above for given estimates $n_j$ of the frequencies (if such estimates are available), or even a more complex algorithm that learns the distribution of the queries and the corresponding optimal allocation over time.

Define the *adjusted discount function* $\Phi_\alpha(f) = 1 - e^{\alpha\cdot(f-1)}$, where $f \in [0,1]$ denotes the fraction of the budget spent by an advertiser. The parameter $\alpha$ controls the extent to which the algorithm relies on $O$. As a new query for keyword $j$ arrives, the algorithm finds advertiser $p$ that maximizes $\Phi(f_i)b_{ij}$ over all $i \in A$. Also, $O$ recommends advertiser $o$ to receive the query. The algorithm compares $\alpha.\Phi(f_o)b_{oj}$ with $\Phi(f_p)b_{pj}$. If the former is bigger than the latter, then the query is allocated to $o$; otherwise, it is allocated to $p$. The algorithm can be presented as the following:

---

Algorithm Q($\alpha$)

---

**Upon the arrival of a new query for keyword** $j \in K$**:**
  Let $o \in A$ be the advertiser recommended by $O$ for receiving $j$.
  Let $p \in A$ be the advertiser with maximum $\Phi_\alpha(f_i)b_{ij}$ among all $i \in A$.
**if** $\alpha\Phi_\alpha(f_o)b_{oj} \geq \Phi_\alpha(f_p)b_{pj}$ **then**
  Allocate $j$ to $o$.
**else**
  Allocate $j$ to $p$.
**end if**

---

# 4 Empirical Study

## 4.1 Implementing Algorithms

The first part of Algorithm $Q(\alpha)$ is the optimistic algorithm $O$. This algorithm uses offline optimization with estimates of queries' frequency to allocate ads in an online manner. The offline component solves a linear program to determine an optimal allocation of queries to advertisers based on predicted query volumes and bid values, subject to budget and capacity constraints. The LP solution is then rounded to integer allocations using a priority mechanism that respects constraints. During the online phase, queries arrive sequentially, and the algorithm uses the offline allocation results as guidance to assign each query to the advertiser with the highest feasible bid, ensuring budget compliance. As discussed by Mahdian et al. (2012), If we see a frequency more than what was predicted, the expert simply discards the query. Also, when a query of a certain type arrives, we allocate the query to the highest bid among the advertisers who are assigned this query in the offline solution.

The second part of $Q(\alpha)$ is what we call pessimistic algorithm $P$. It is based on Mehta et al. (2007) with a parameter $\alpha$, and simply recommends the advertiser with maximum $\Phi_\alpha(f_i)b_{ij}$ among all $i \in A$.

Now we can implement algorithm $Q(\alpha)$, that we call balanced algorithm. The balanced algorithm integrates the optimistic and pessimistic allocation to dynamically decide query allocations in an online setting. For each query, the algorithm evaluates recommendations from both strategies using the $\Phi_\alpha$ function, which combines the advertiser's remaining budget fraction and bid amount. The final allocation is determined based on the scores, with the optimistic score weighted by the parameter $\alpha$.

## 4.2 Dataset Generation

An input generator module is designed to generate inputs for testing the performance of the optimistic, pessimistic, and balanced algorithms in online query allocation experiments. It creates a certain number of advertisers with random budgets and bids influenced by noise for variability. Advertisers' budgets are generated within a specified range, while bids are created for multiple keywords, also incorporating randomness. Additionally, the module generates a random sequence of queries. These inputs—advertisers, bids, and query sequences—are bundled together for use in algorithm testing.

There is a functionality in the input generation that is designed to simulate prediction inaccuracies in query distributions by introducing controlled deviations from an actual query sequence. Given the actual sequence, the number of distinct keywords, and using a specified *prediction error level*, the function modifies the original distribution to reflect a degree of error proportional to the specified level. When the error level is zero, the function accurately returns the original query sequence. For non-zero error levels, it adjusts keyword frequencies by redistributing occurrences among keywords, thereby simulating prediction errors while maintaining the total query count. This method is integral to evaluating algorithm performance under varying levels of prediction uncertainty, a critical consideration in online ad allocation.

## 4.3 Experimental Setup

The experiment evaluates the performance of three ad allocation algorithms: optimistic, pessimistic, and balanced, under varying conditions. Inputs, consisting of advertisers, bids, and query sequences, are generated using the input generator module. The experiment is designed to assess the revenue generated by each algorithm across multiple replications, with different levels of prediction error in query frequencies and a range of $\alpha$ values to control the balance in the balanced algorithm. Each algorithm processes the same query sequence independently, resetting budgets and advertiser states before evaluation. The results, including revenues from all algorithms, are aggregated across replications and stored for further analysis. This framework allows a systematic comparison of the algorithms' performance under diverse scenarios.

# 5 Results

The experiment evaluates the performance of three ad allocation algorithms—Optimistic, Pessimistic, and Balanced—under varying levels of prediction error and a balancing parameter, $\alpha$. The setup involves a simulated environment with 100 advertisers, 100 keywords, and a query sequence of 1,000 queries. Advertisers are initialized with random budgets, and their bids are generated within a specified range. The algorithms process the queries and allocate advertisers based on their respective strategies.

To assess the impact of prediction error and $\alpha$, the experiment is replicated 10 times for three prediction error levels (0.0, 0.25, 0.5) and five $\alpha$ values (1.0, 2.0, 4.0, 5.0, 10.0). Each algorithm's revenue is recorded and averaged across replications for each combination of parameters. The resulting data is visualized through line plots, with separate graphs for each prediction error level. These plots highlight trends in revenue for each algorithm as $\alpha$ varies, offering insights into their performance under different uncertainty levels and parameter settings. Figures 1, 2, and 3 show the comparison of total revenue across algorithms with different levels of prediction accuracy.

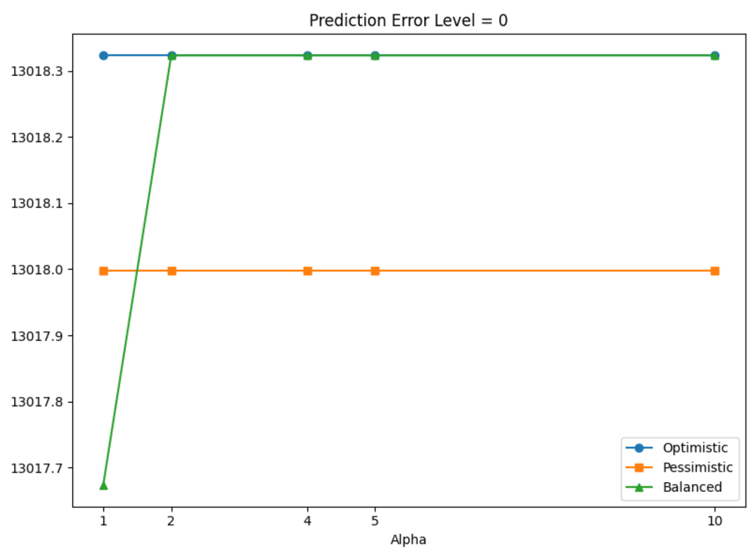


Figure 1: Total revenue across algorithms when prediction error is zero.

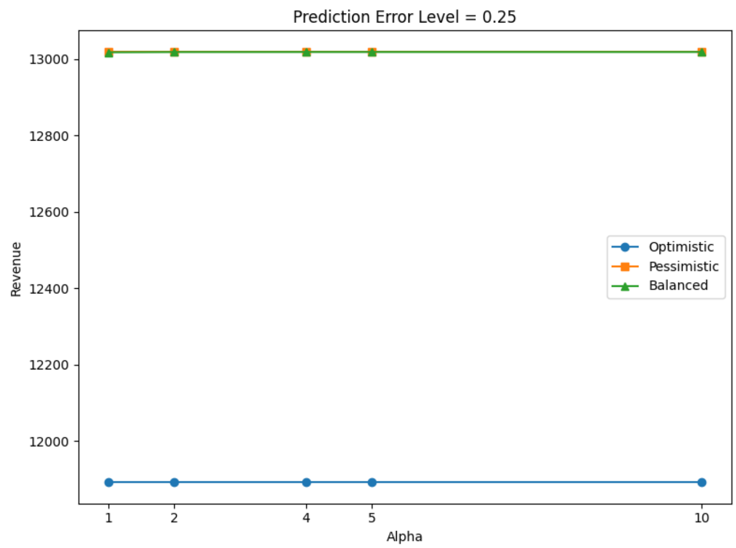


Figure 2: Total revenue across algorithms when prediction error is 25%.

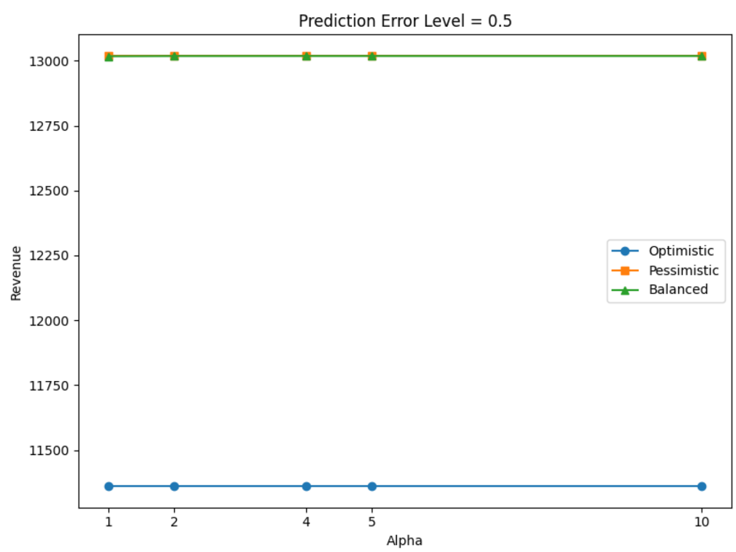


Figure 3: Total revenue across algorithms when prediction error is 50%.

As shown in the figures, the performance of the optimistic algorithm is higher than others when the predictions are accurate, but it drops as the prediction error increases. The pessimistic algorithm performs pretty much the same for different sets of inputs, which is what we expect since it has a theoretical competitive ratio guarantee. For the balanced algorithm introduced by Mahdian et al. (2012), we can see it matches the performance of the better algorithm in most cases. The balanced algorithm demonstrates a unique capability to leverage the strengths of both optimistic and pessimistic approaches. The algorithm can effectively capture potential revenue from accurate predictions while maintaining a robust mechanism to handle prediction errors.

# 6 Discussion

The experimental findings contribute insights into online optimization problems with uncertain information. The balanced algorithm proposed by Mahdian et al. (2012) demonstrates noteworthy performance characteristics. Its behavior across different values of the hyperparameter $\alpha$ suggests an inherent adaptability that mitigates the need for precise parameter tuning.

The algorithm's ability to efficiently capture accurate predictions while maintaining resilience to prediction errors is particularly significant. In practical scenarios characterized by inherent input uncertainty, the algorithm's integrated approach of combining optimistic and pessimistic strategies enables dynamic allocation. This approach effectively reduces the risks associated with purely optimistic or pessimistic methods by providing a more nuanced allocation mechanism.

Future research directions could explore several promising avenues. First, extending the algorithm to more complex domains with additional constraints, such as multi-objective optimization or scenarios with time-varying bid structures, could provide deeper insights into its generalizability. Investigating the algorithm's performance under more sophisticated prediction error models or in domains beyond online advertising, such as resource allocation in cloud computing or dynamic pricing, would further validate its potential. Additionally, developing machine learning techniques to adaptively select the $\alpha$ parameter based on historical performance could enhance the algorithm's autonomous adaptation capabilities.

The implementation of this approach, along with the experimental code, can be found in the project's GitHub repository: `https://github.com/RohamBahri/online_query_allocation`.

# References


[1] M. Mahdian, H. Nazerzadeh, and A. Saberi. 2012. Online optimization with uncertain information. *ACM Trans. Algorithms* 8, 1 (Jan. 2012), 2:1–2:29.

[2] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. 2007. Adwords and generalized online matching. *J. ACM* 54, 5 (Nov. 2007), 1–13.