View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by AIS Electronic Library (AISeL)

Online Auctions with Dual-Threshold Algorithms: An Experimental Study and Practical Evaluation

Philipp Knöpfle[1], Johannes Knörr[1], Sören Merting[2]

[1] Technical University of Munich, Department of Informatics, Munich, Germany
{philipp.knoepfle,johannes.knoerr}@tum.de
[2] Technical University of Munich, Department of Informatics, Munich, Germany
{soeren.merting}@in.tum.de

Abstract. Online auctions are a viable alternative to conventional posted price mechanisms. Agrawal, Wang, and Ye [1] have proposed two primal-dual algorithms for revenue-maximizing multi-item allocation tasks. Although promising in terms of theoretical properties and competitive ratios, there is a lack of evidence regarding the real-world practicability of these mechanisms, for instance referring to online auction-based tickets sales. In this paper, we conduct an experimental study on both the *One-Time Learning Algorithm* (OLA) and the *Dynamic Learning Algorithm* (DLA) based on synthetic data, revealing the remarkable aptitude of the latter for non-trivial online auctions. Being robust to most input variations, the inherent dynamic update of dual thresholds achieves a superior balance with respect to the trade-off between objective function values and runtimes. We address critical sensitivities quantitatively and draft several small extensions by incorporating input distribution knowledge.

Keywords: Online Auctions, Online Ticket Sales, Experimental Study

1 Introduction

Online auctions and auction-type mechanisms become increasingly popular for revenue-maximizing allocations of scarce resources. While display ad auctions and sponsored search are always based on bidders revealing their willingness-to-pay prior to the actual allocation, many B2C business models still rely on conventional posted prices. For instance, ticket sales for cultural events or sports competitions are often conducted with a fixed-price policy, occasionally replaced by concepts of revenue management or dynamic pricing. In this case, however, sellers have little knowledge on the willingness-to-pay or consumer surplus of the bidders, potentially preventing higher revenues. In contrast, auction-type allocations in an online fashion can enable a better absorption of buying power and hence a more efficient allocation. Recently, in the aviation or hotel industry it can be observed that companies experiment with their established pricing models by gradually substituting conventional posted price

14th International Conference on Wirtschaftsinformatik,
February 24-27, 2019, Siegen, Germany

mechanisms with online open auctions. Such auctions might also be of particular interest for ticket sales in order to reduce black market activity. Aside from the online arrival of bidders, however, ticket allocation problems often exhibit additional complexity as a result of excess demand, heterogeneous willingness-to-pay and short processing times, calling for performant and fast decision-making algorithms.

We examine a notable contribution by [1] in the context of online resource allocation. Based on a primal-dual approach, the authors state theoretical properties and postulate a broad applicability of their two algorithms vis-a-vis auction-type allocations. As with many primal-dual frameworks, however, little is known about the implementation, real-world feasibility or practical challenges of these mechanisms.

This paper seeks to examine this gap between theory and practicability by assessing the empirical applicability of both the *One-Time Learning Algorithm* (OLA), which calculates a single set of dual threshold prices, and the *Dynamic Learning Algorithm* (DLA), which continuously updates dual prices at geometric time intervals. Optimizing the revenue against a stationary bidding process, we perform numerical experiments on both algorithms based on synthetic data and compare the runtimes and objective values to three benchmarks. Furthermore, we examine reasonable parameter combinations and investigate the algorithms' sensitivity with respect to their input parameters. We find a strong trade-off between outcome quality and runtime, whereas the DLA produces superior outputs while maintaining moderate runtimes for almost all cases. Our experiments provide evidence that the DLA is especially suited for complex allocation tasks with resource scarcity, heterogeneous bidders and limited ex-ante knowledge. Moreover, addressing the most critical sensitivity, we suggest several extensions utilizing priorly known input distribution information and thus improving the practical applicability for online auctions.

The remainder of this paper is structured as follows. Section 2 begins by reviewing recent contributions in the online resource allocation literature. We also explain the algorithms of [1] in more detail. We present our implementation and experimental design in section 3.1. Section 3.2 provides the experimental observations and states key results. We discuss our findings with respect to extensions and put emphasis on enhancing the practicability. Section 4 concludes and suggests further research.

2 Review of Literature and Research

Optimizing the allocation of given resources with respect to revenue or social welfare is a core element of a variety of scientific contributions. A popular approach is to actively control the availability of resources, often referred to as revenue management [16]. Closely related is the concept of dynamic pricing, in particular dynamically adjusting posted prices for the purpose of revenue maximization. Comprehensive literature overviews for value-based pricing techniques are, for instance, provided by [12] or [16]. Although these concepts ultimately aim at exploiting differences in customer willingness-to-pay, one key drawback of such techniques is the failure to explicitly record willingness-to-pay and hereby enhance the producer surplus.

Another domain of resource allocation is auction theory. While other objectives such as efficiency or states of equilibria can certainly exist, revenue maximizations might also be pursued through auctions. Since bidders reveal their true willingness-to-pay in incentive-compatible mechanisms, sellers might be able to absorb a larger proportion of the consumer surplus, thus increasing revenue. Traditional auction theory usually refers to a single point in time with comprehensive knowledge of demand and supply. However, an increasing number of digital markets are dynamic, meaning that bidders reveal their willingness-to-pay sequentially, requiring an immediate allocation decision by the auctioneer. The growing field of online auctions focuses – amongst others – on the algorithmic design for this class of problems. Examples include, for instance, [2], [14], and [17]. [13] specifically address the case of online ticket sales and design a fully stochastic and dynamic algorithm in order to compute an optimal online auction mechanism. [9] demonstrate that posted price mechanisms can indeed match competitive ratios of combinatorial auction principles, provided that stochastic information of the bidders' valuations is available.

Among a variety of algorithmic frameworks, one popular approach for online allocations is the utilization of dual threshold prices. Reasons for this include, for instance, a wide field of application, simple interpretability as well as the ease of implementation. In essence, this concept makes use of shadow prices of an ex-ante linear program as weights for subsequent decision-making. An incoming request can thus only be accepted if the weighted resource consumption is exceeded by the communicated willingness-to-pay. For instance, [8] investigate the AdWords problem within a similar framework as [1]. Under a random permutation assumption with known number of bidders and a specific right-hand side condition, they retrieve a $(1-\varepsilon)$-competitive algorithm. In a similar manner, but assuming an i.i.d. input with unknown and changing distributions, [7] develops dual-based resource allocation algorithms for stochastic AdWords problems. In a series of interrelated contributions, [3-5] use primal-dual approaches to match and prove several competitive ratios for various kinds of online problems, for example ad-auctions. Other examples using dual approaches include [10-11] for display ads or [6] in a Bayesian auction setting.

However, some contributions, for example those relating to the AdWords problem, do not permit multidimensional demand vectors, in particular the possibility to request several resources simultaneously, and are thus not directly applicable to ticket sales or similar B2C businesses. In contrast, the algorithms proposed by [1] explicitly refer to classical multi-item revenue maximization problems. The authors also demonstrate a superiority with respect to the theoretical competitive ratios compared to related online auction frameworks.

In order to allocate resources in an online auction fashion [1] use online linear programming, where the constraint matrix and corresponding objective function coefficients are revealed column by column over time. The linear program is calculated based on the input received so far and without any information about future requests. However, all subsequent decisions are based on this solution. More precisely, consider the following linear program:

$$\begin{aligned}
&\text{maximize } \textstyle\sum_{j=1}^{n} \pi_j x_j && \\
&\text{subject to } \textstyle\sum_{j=1}^{n} a_{ij} x_j \leq b_i && i = 1, \dots, m \\
&x_j \in [0,1] && j = 1, \dots, n,
\end{aligned} \tag{1}$$

where m denotes the number of capacity constraints and n represents the number of columns. While a_{ij} stands for the requested items by bidder j regarding resource i with capacity b_i, the term π_j denotes the willingness-to-pay for the total package. In an online auction the coefficients $(\pi_j, \boldsymbol{a_j})$ are revealed consecutively over time. The contribution of [1] is to compute the dual solution of a partial linear program and use it as a decision rule for future incoming bidders. The key idea is to set threshold prices $\boldsymbol{p} = (p_i)_{i=1,..,m}$ for each resource i equal to the dual prices of a linear program that is solved after a fraction of $\varepsilon \in (0,1)$ columns are revealed. Subsequent incoming bids $(\pi_j, \boldsymbol{a_j})$ will be compared to the current threshold prices $\boldsymbol{p}$. A bid will only be accepted if π_j exceeds $\boldsymbol{p^T a_j}$ and if no resource constraint is violated, meaning the problem remains feasible with $x_j = 1$.

OLA:	DLA:	(2)
maximize $\sum_{j=1}^{s} \pi_j x_j$	maximize $\sum_{j=1}^{l} \pi_j x_j$	
s.t. $\sum_{j=1}^{s} a_{ij} x_j \leq (1-\varepsilon)\frac{s}{n} b_i$	s.t. $\sum_{j=1}^{l} a_{ij} x_j \leq (1-h_l)\frac{l}{n} b_i$	$i = 1, \dots, m$
$x_j \in [0,1]$	$x_j \in [0,1]$	$j = 1, \dots, s$

[1] present two algorithms: the so called *One-Time Learning Algorithm* (OLA) and the *Dynamic Learning Algorithm* (DLA). The respective partial linear programs for retrieving the sets of dual prices are denoted above. The OLA learns a single threshold price vector at time $s = \varepsilon n$, applicable to all following bids. All incoming requests until s are only used to calculate the threshold prices and will always be rejected. Assuming that n is known and $(\pi_j, \boldsymbol{a_j})$ arrive in random order, [1] show that the OLA exhibits $(1-6\varepsilon)$-competitiveness against the ex-post optimum (OPT) under the right-hand side condition $B = \min_i b_i \geq \frac{6m\log(n/\varepsilon)}{\varepsilon^3}$.

The DLA updates dual prices in geometric intervals at $\varepsilon n, 2\varepsilon n, 4\varepsilon n, 8\varepsilon n, \ldots$, in particular for each $l = \lceil 2^r \varepsilon n \rceil$ with the largest integer r such that $l < j$. The right-hand side of the partial linear programs is modified by a factor $h_l = \varepsilon\sqrt{\frac{n}{l}}$. The DLA is $1 - O(\varepsilon)$ competitive given a milder condition $B = \min_i b_i \geq \frac{m\log(n/\varepsilon)}{\varepsilon^2}$.

Note that both algorithms are distribution-free. However, [1] need ex-ante knowledge about the size of n in order to calculate the learning fractions of the respective algorithm. Furthermore, [1] assume that the bids $(\pi_j, \boldsymbol{a_j})$ arrive in random order. The latter assumption seems reasonable from a practical perspective, as the order of columns usually appears to be independent of the columns' content.

3 Experimental Analysis

3.1 Experimental Design

The main contribution of [1] is the introduction and (theoretical) analysis of these two algorithms for solving online allocation problems. However, they do not feature any specific implementation or application. This paper intends to examine the implied allocation mechanism from a practical perspective and to analyze sensitivities.

For this purpose, the algorithms are implemented and systematically tested in numerical experiments. In particular, all input parameters are varied in an orderly fashion to isolate important influencing factors on the algorithms' outcome and runtime. The treatment variables include the number of resources m, resource capacities b, initial learning fraction ε as well as number of bidders n. Moreover, building on the distribution-free property of the algorithms, we examine the robustness with respect to different stationary input distributions or distribution parameters regarding the willingness-to-pay π_j and the item requests a_{ij}.

Table 1. Parameters and Treatment Variables

	Description	*Values and Distributions*
Parameters	Simulations per Configuration	10
	Permutations per Simulation	100
	$\max(a_{ij})$	5
	a_{1j}, a_{2j}, a_{3j}	$N(0.4, 0.3), N(0.5, 0.6), N(0.6, 0.9)$
	$\pi_{1j}, \pi_{2j}, \pi_{3j}$	$N(100, 30), N(75, 20), N(50, 10)$
Treatment Variables	Resources m	$\{1; 2; \mathbf{3}; 5; 10\}$
	Capacity b	$\{50; 100; \mathbf{200}; 300; 400; 500; 600; 700; 800; 900; 1{,}000\}$
	Fraction ε	$\{0.001; \mathbf{0.01}; 0.02; 0.03; 0.04; 0.05; 0.075; 0.1; 0.125; 0.15; 0.2; 0.25\}$
	Bidders n	$\{200; 400; 600; 800; \mathbf{1{,}000}; 1{,}200; 1{,}400; 1{,}600; 1{,}800; 2{,}000; 5{,}000; 10{,}000\}$

The treatments are analyzed with respect to runtimes as well as objective function values, i.e. the revenue generated from the allocation in percent of the ex-post optimum. We define a baseline treatment which serves as a reference point for the subsequent ceteris paribus variation of treatment variables. The values and ranges for each parameter and variable are summarized in Table 1. The respective baseline configuration ($m = 3$, $b = 200$, $\varepsilon = 0.01$, $n = 1{,}000$) is highlighted in bold characters. All resources are assumed to have an identical capacity $b_i = b$. The parameter a_{ij} is required to be an integer value between zero and a pre-specified upper limit. In order to map distinct resource classes, the parameter a_{ij} follows a normal distribution with stepwise mean-variance-tuples. Similarly, the willingness-to-pay decrease over the different resources and are summed up to an aggregate π_j. Every bidder j is obligated to request at least one item.

Each treatment variable is considered in an isolated fashion. Under the ceteris paribus assumption, each input parameter is varied according to Table 1 using the baseline configuration as a starting point. Hence, our experimental design comprises 40 treatment combinations. For each configuration, 10 instances are created. Since [1] consider expected values over all permutations, each instance consists of 100 sub-instances, representing different permutations of the set of bidders.

Regarding the runtime evaluation, it should be noted that single permutations might be subject to noise distorting the measured runtimes. However, since averages are taken over a multitude of permutations, valid statements on general trends are ensured. Furthermore, we only intend to examine diverging runtime magnitudes and to identify common patterns. In order to explicitly exclude the possibility of invalid runtimes, we checked the results against larger problem instances. Essentially, we scaled the test configurations up by a factor of 100 (e.g. $b = 20{,}000$, $n = 100{,}000$ with 10 permutations and 5 simulations as the baseline treatment) and computed the respective ratios of runtimes. These ratios also took values of the up-scaling factor of 100, which we found to be consistent with our previous runtime analysis. Absolute runtimes should, however, always be handled with care according to [15].

In order to evaluate the objective function values and runtimes of both the OLA and the DLA, several simple benchmarks have been implemented as alternative measures. The most simplistic mechanism would be to accept any incoming request, as long as no constraint is violated. This quick and easy allocation principle is referred to as *Greedy Algorithm* in our context. In contrast, a so-called *Interval Learner* updates dual prices at constant 10%-intervals with respect to the ex-ante known number of incoming bids n. This benchmark is expected to always exhibit longer runtimes. The last benchmark, the *Willingness-To-Pay (WTP) Learner,* is based on the idea that it might be reasonable to update dual prices whenever the average over all bids per item changes by a certain magnitude, in our case by at least 5% in absolute terms. That is, if the pool of bidders appears to become more heterogeneous, different threshold prices might be deemed appropriate, calling for a re-calibration of dual prices. A fraction of 10% of the bids is used to calculate the initial dual prices.

The simulations were implemented in Python 3.6. Linear programs were solved using the interface to Gurobi 8.0. All simulations were performed on an Intel Core i7 7700K 4.20GHz quad-core machine with 32 GB of RAM.

3.2 Results

Starting with the baseline treatment, the well-known trade-off between approximation capabilities regarding the OPT and average runtimes already becomes visible. The numerical results indicate that the DLA excels in terms of the objective function (93.68% of the OPT). Evidently, the frequent update of dual prices incorporated into the DLA generates considerably better objective function values as opposed to the simpler OLA (69.42%), at least given the input parameters at hand. Furthermore, in spite of several updating steps of dual prices, the DLA certainly seems to be able to balance objective function output and runtime ($126.47ms$; OLA: $8.36ms$) reasonably. In contrast, while the *Greedy Algorithm* (80.27%; $1.53ms$) and the *WTP Learner* (86.25%; $97.31ms$) are executed faster, they cannot provide comparable approximations of the OPT due to their naivety. Likewise, the *Interval Learner* may come closer to the DLA vis-à-vis the objective (88.96%), but requires significantly more runtime ($405.35ms$). The OLA only proves to be competitive for large problem instances. Most notably, its approximation ratio increases to 95.16% in the up-scaled control scenario, potentially because the number of bids εn used for learning the dual prices is bigger in absolute terms. Therefore, the calibrated thresholds might be more valid than in the small-scale case.

Result 1. *The DLA provides a superior balance between objective function values and average runtimes. While several re-calibrations of dual thresholds enable good approximations of the ex-post optimum, the geometric updating intervals shift the majority of computational intensity toward small-scale optimization problems.*

Under the ceteris paribus condition, a variation of the total number of resources m, each having a capacity of 200 units, does not significantly affect the approximation performance of either the OLA or the DLA, as can be seen in Table 2. Employing a very little or a very large number of resources does not deteriorate the outcome substantially, potentially since each added resource inevitably comes along with new demand, as guaranteed by the simulation specifics. The performance of the *Greedy Algorithm*, however, constantly drops with an increasing number of resources. Table 2 also reports linear regression slope coefficients and adjusted R^2 figures for the objective function and the runtime. Regarding these coefficients, note that the dependent variables are denoted in percent and milliseconds (ms), respectively. Moreover, a significance level of 1% is chosen. The negative linear objective function sensitivity of the *Greedy Algorithm* may be ascribed to growing heterogeneity among bidders as a result of more resources and hence more scope of simulation. Since more deliberate decisions are necessary for a diverse pool of bidders, the naïve *Greedy Algorithm* cannot sustain its approximation ratio. The runtimes for almost all mechanisms increase significantly and in a linear fashion once new resources are added.

Because an increase in the number of resources goes hand in hand with newly generated demand, resource scarcity or abundance can better be reflected by changing the available capacities b. As displayed in Table 3, the *Greedy Algorithm*, the DLA, and the OLA ultimately converge to the OPT in terms of the objective function value when resources are excessively available. In particular, a 100%-approximation of the *Greedy Algorithm* indicates that all incoming bidders can be served. In this case, shadow prices may be close to zero and the other mechanisms exhibit a gap to the OPT mainly due to the initial calibration period, where all requests are rejected. When resources are scarce, however, the algorithms show significant discrepancies. As items need to be assigned with consideration, simple allocation mechanisms, as implied by the OLA or the *Greedy Algorithm*, produce below average results. The OLA, however, exhibits the steepest linear growth with increasing resource capacity. In contrast, the DLA already performs very well for limited availabilities. It also exhibits a slightly quadratic relationship, i.e. performing a quadratic regression increases the $\bar{R}^2$ from 86.45% in the linear case to 96.49% with the quadratic regression coefficient being statistically significant at the 1% level. That is, the DLA dominates all other benchmarks as long as resources are exposed to scarcity to some extent. It also exceeds the *WTP Learner* and the *Interval Learner*, where the first 10% of the bidders will always be rejected. Changing b does not significantly affect the average runtimes aside from the OLA and the *Greedy Algorithm*. That is, if more resource capacities are available, more bidders can be served, leading to a consistent upward trend in runtime, albeit on a small level. For the other mechanisms, this effect does not become visible, as it is only a tiny proportion of the total runtime.

The fraction ε represents an interesting lever for training-based algorithms, determining the number of bids initially required for calibrating the thresholds. As can be seen in Table 4, the three benchmarks exhibit zero sensitivity, as they do not make use of this parameter. In terms of the objective function value, the DLA shows the greatest dependence with respect to ε. Generally speaking, the smaller this fraction is chosen, the more learning instances are executed by the algorithm, enabling a better approximation of the OPT. For large ε, the OLA produces better results than the dynamic mechanism, indicating that the DLA is considerably restricted by the modifying right-hand side factor postulated by [1], artificially increasing the dual prices at each re-calibration for too large fractions. At the same time, unlike the DLA exhibiting a linearly decreasing behavior, the OLA produces its best results for a medium $\varepsilon = 0.05$. Since it only learns dual prices at a single time, a very small fraction of the sample will not be representative enough and thus result in a poor or incalculable performance. Therefore, in order to produce a proper outcome, a certain minimum share of bids needs to constitute the training set. The OLA thus exhibits a significant quadratic relationship with an $\bar{R}^2$ of 85.69%. Furthermore, the tests reveal linearly increasing OLA runtimes, since the single optimization problems will encompass more elements with increasing ε. As evident from the numerical results, the shape of the DLA runtime function is rather serrated. Holding the total number of required optimization steps constant, the runtime would increase with growing ε for the same reasons as the OLA. However, once ε exceeds some threshold, one former optimization step is not feasible anymore, thereby reducing the total number of dual price updates and significantly decreasing the average runtime.

Table 2. Numerical Results for Treatment Variable m

Resources m	*1*	*2*	*3*	*5*	*10*	β	$\bar{R}^2$
DLA	91.06%	93.64%	93.77%	92.49%	88.69%	-	-
	118.20ms	165.96ms	209.76ms	291.21ms	503.05ms	42.43	99.96%
OLA	70.21%	71.20%	69.24%	69.27%	65.08%	-	-
	9.49ms	12.78ms	14.45ms	18.05ms	27.70ms	1.95	99.31%
Greedy	87.65%	86.22%	79.78%	72.96%	62.55%	-2.85	95.19%
Algorithm	1.07ms	2.19ms	2.68ms	3.35ms	5.40ms	0.444	95.57%
Interval	84.24%	88.39%	89.27%	89.50%	87.89%	-	-
Learner	361.09ms	517.65ms	662.04ms	942.33ms	1675.44ms	145.33	99.99%
WTP	83.33%	86.79%	86.09%	85.20%	81.25%	-	-
Learner	42.10ms	146.69ms	160.14ms	221.53ms	439.07ms	40.83	96.52%

Table 3. Numerical Results for Treatment Variable b

Capacities b	*50*	*100*	*200*	*300*	*400*	*500*	*600*	*700*	*800*	*900*	*1,000*	β	$\bar{R}^2$
DLA	89.43%	91.78%	93.73%	94.70%	95.83%	96.52%	97.36%	97.78%	97.77%	98.59%	98.92%	0.00868	86.45%
	225.52ms	221.24ms	222.40ms	221.65ms	222.79ms	223.30ms	223.83ms	224.49ms	227.53ms	226.07ms	223.96ms	-	-
OLA	67.56%	63.57%	70.42%	74.67%	79.92%	83.77%	86.90%	90.32%	92.21%	94.93%	96.96%	0.0351	96.78%
	14.97ms	14.99ms	16.18ms	16.68ms	17.52ms	17.93ms	18.70ms	19.50ms	19.86ms	20.01ms	20.19ms	0.00595	96.72%
Greedy	73.40%	77.87%	79.73%	79.77%	83.10%	83.64%	87.27%	92.26%	95.79%	100.00%	100.00%	0.0278	96.07%
Algorithm	2.54ms	3.08ms	3.59ms	4.31ms	5.00ms	4.84ms	4.97ms	4.73ms	5.54ns	5.22ms	5.46ms	0.00274	77.88%
Interval	86.93%	88.03%	89.04%	89.53%	89.39%	89.47%	89.57%	89.70%	89.72%	89.89%	89.99%	0.00230	61.23%
Learner	710.78ms	696.32ms	693.92ms	689.87ms	691.53ms	689.57ms	690.73ms	693.39ms	701.08ms	690.55ms	686.45ms	-	-
WTP	80.73%	83.43%	86.34%	87.80%	88.23%	88.88%	89.25%	89.61%	89.59%	89.74%	89.99%	0.00784	69.40%
Learner	176.46ms	173.45ms	174.65ms	170.45ms	175.35ms	169.06ms	172.63ms	178.59ms	176.45ms	170.80ms	172.75ms	-	-

Table 4. Numerical Results for Treatment Variable ε

Fraction ε	*0.001*	*0.01*	*0.02*	*0.03*	*0.04*	*0.05*	*0.075*	*0.1*	*0.125*	*0.15*	*0.2*	*0.25*	β	$\bar{R}^2$
DLA	94.54%	93.67%	92.48%	90.06%	88.53%	86.32%	80.84%	75.88%	70.36%	65.16%	56.39%	47.25%	-196.66	99.82%
	109.82ms	125.89ms	123.25ms	176.91ms	120.06ms	146.36ms	108.35ms	140.16ms	85.01ms	99.91ms	129.10ms	72.08ms	-	-
OLA	76.12%	70.19%	76.47%	78.81%	80.47%	81.49%	80.87%	79.79%	76.26%	73.28%	65.87%	58.22%	-62.73	45.92%
	8.35ms	8.26ms	9.12ms	9.81ms	10.63ms	11.46ms	13.38ms	15.33ms	17.25ms	19.23ms	23.20ms	27.09ms	77.39	99.87%
Greedy	80.67%	79.71%	80.26%	79.61%	79.88%	80.13%	79.91%	80.85%	79.14%	80.08%	80.02%	80.11%	-	-
Algorithm	1.54ms	1.54ms	1.53ms	1.54ms	1.54ms	1.51ms	1.53ms	1.56ms	1.54ms	1.54ms	1.54ms	1.52ms	-	-
Interval	88.97%	88.98%	89.15%	88.76%	88.95%	89.07%	89.06%	89.10%	89.10%	88.82%	88.91%	88.83%	-	-
Learner	404.87ms	403.81ms	403.80%	404.97ms	403.68ms	404.70ms	403.79ms	404.57ms	403.14ms	404.58ms	404.65ms	404.70ms	-	-
WTP	86.28%	86.39%	86.36%	86.11%	86.14%	86.45%	86.21%	86.48%	86.16%	86.14%	86.38%	86.32%	-	-
Learner	101.45ms	100.35ms	98.78ms	97.56ms	99.01ms	98.01ms	99.38ms	97.96ms	97.67ms	99.96ms	99.44ms	97.07ms	-	-

Table 5. Numerical Results for Treatment Variable n

Bidders n	*200*	*400*	*600*	*800*	*1,000*	*1,200*	*1,400*	*1,600*	*1,800*	*2,000*	*5,000*	*10,000*	β	$\bar{R}^2$
DLA	98.36%	95.30%	93.88%	93.64%	93.64%	93.56%	93.45%	93.18%	93.48%	93.42%	92.98%	92.54%	-	-
	29.96ms	54.14ms	78.36ms	102.48ms	126.14ms	150.63ms	174.37ms	198.78ms	222.92ms	247.69ms	607.05ms	1208.25ms	0.120	100.00%
OLA	86.91%	77.39%	71.91%	70.33%	69.62%	68.74%	69.36%	69.32%	69.11%	69.22%	69.88%	69.83%	-	-
	2.89ms	4.18ms	5.58ms	6.94ms	8.24ms	9.56ms	10.91ms	12.19ms	13.48ms	14.83ms	33.95ms	65.72ms	0.00640	99.99%
Greedy	100.00%	84.44%	81.01%	79.57%	79.84%	80.24%	81.25%	80.10%	78.94%	78.33%	71.80%	68.25%	-0.00194	45.70%
Algorithm	0.47ms	0.82ms	1.09ms	1.33ms	1.55ms	1.74ms	1.95ms	2.14ms	2.32ms	2.50ms	5.03ms	9.25ms	0.00087	99.64%
Interval	89.83%	89.23%	89.14%	89.11%	89.10%	88.75%	88.83%	88.35%	88.59%	88.44%	87.94%	87.15%	-0.00023	79.86%
Learner	84.96ms	166.57ms	246.56ms	325.99ms	404.26ms	484.82ms	562.31ms	641.81ms	720.15ms	800.69ms	1987.33ms	3973.44ms	0.396	100.00%
WTP	89.73%	87.88%	87.09%	86.59%	86.48%	85.83%	85.94%	85.23%	85.53%	85.24%	84.52%	84.32%	-	-
Learner	29.51ms	49.00ms	65.75ms	82.25ms	97.93ms	114.43ms	128.86ms	150.19ms	163.79ms	182.18ms	419.64ms	828.72ms	0.0811	99.99%

Changing the number of stationary bidders n competing for the fixed resource capacities, the DLA once again proves to be robust. This can be seen in Table 5. While it becomes increasingly difficult to select the ex-post optimal requests from a larger set of bidders, the drop in performance is not as significant as with other benchmarks. The *Greedy Algorithm*, for instance, cannot maintain its approximation ratio. The *WTP Learner* and the *Interval Learner* remain robust to a certain extent, yet never reaching the performance of the DLA. The OLA, in turn, is more sensitive towards changes in n. With a growing number of requests, a single learning step is not sufficient due to the multitude of bidders arriving subsequent to the calibration. The average runtimes of all algorithms naturally increase with problem size. The functional relationship concerning the runtimes is linear for all mechanisms.

We validated the observations of these ceteris paribus analyses with a multivariate parameter grid and were not able to detect major deviations. The results above might, however, also be contingent upon the distribution assumptions. In general, there are six levers to be tested with respect to the stationary input processes: mean and standard deviation of the willingness-to-pay π_j and of the requests a_{ij}, given a normality assumption, as well as the underlying distributions of the simulated parameters themselves. As pointed out by [1], the DLA stands out due to its distribution-free property and thus robustness to all tuning parameters. Conversely, the *Greedy Algorithm* benefits from homogeneous pools of bidders, exhibiting major improvements for decreasing standard deviations of π_j. It is also most sensitive to changes of the underlying distributions of π_j and a_{ij}. The normality assumption appears to be most suitable for achieving little runtimes. If the distribution parameters fluctuate within a simulation run, the DLAs performance deteriorates, yet is still competitive to the benchmarks.

Result 2. *The DLA proves to be robust regarding changes of most input parameters. Its capability to produce near-optimal objective function values remains widely unaffected by changes in the number of resources, available resource capacities, number of bidders, or distribution assumptions. The DLA thus seems to be applicable to any kind of online auction configuration with a revenue maximization objective.*

Result 3. *The average runtimes of the DLA depend linearly on the number of resources and bidders. They are stable with respect to resource capacities. These statements also apply to the OLA and more naïve benchmarks.*

Result 4. *The initial learning fraction ε represents the most critical sensitivity for the DLA. Small fractions are necessary in terms of the objective function value, but come at the cost of more computing time. The indispensable refusal of the first εn bidders is a key drawback with respect to the practical applicability of the DLA.*

Given these experimental results, several interesting inferences can be drawn with respect to the practical applicability of the DLA. In terms of the approximation of the OPT, the findings above indicate that large capacities b, small fractions ε, reasonable numbers of resources m, and few bidders n favor the DLA. The positive discrepancy towards other mechanisms, however, seems to increase for smaller b and larger n, i.e. conditions of resource scarcity and excess demand. Customer heterogeneity, implemented through larger standard deviations for π_j, also contributes to the superiority of this mechanism. In order to validate these statements, a modified baseline treatment is defined with the main purpose of making the DLA the most dominant algorithm. Here, excess demand ($n = 2{,}000$) together with a little learning fraction ($\varepsilon = 0.001$) and more heterogeneous bidders ($\sigma_{\pi_{ij}} = (60,40,20)\ \forall j$) are selected, while all other input figures remain identical to the initial baseline treatment. In this configuration, the DLA distinctly outperforms all other benchmarks with a 94.60%-approximation of the OPT. The *Interval Learner* follows with a ratio of 89.10%, but with more than thrice the runtime of the DLA. While the *WTP Learner* reaches 86.42%, the *Greedy Algorithm* and the OLA only produce 64.56% and 64.51%, respectively. Hence, the DLA excels in terms of OPT-approximation when the allocation is complex, e.g. if resources are scarce and bidders heterogeneous, and thus complies well with the nature of the generic online tickets sales problem. The average runtimes do not differ from our initial findings.

Result 5. *The DLA is especially suited for non-trivial online auction problems, in particular for scenarios with resource scarcity and excess demand by heterogeneous bidders. Its degree of optimality is highly contingent upon the choice of the initial learning fraction ε. Subject to these limitations, the exemplary case of online ticket sales appears to be a viable and expedient area of application.*

The key drawback of the DLA seems to be the unconditional refusal of the first εn bidders, as this fraction is required to learn the first set of threshold prices. Especially for large ε, it might be advantageous to define initial threshold prices to not lose revenue from the learning fraction. Moreover, the right-hand side modifier introduced by [1] artificially increases the dual prices, leading to overly restrictive thresholds for very large ε. Early-arriving bidders might also feel discriminated due to their arbitrary rejection. Since limited computing power or other conceivable reasons might prohibit the employment of sufficiently small learning fractions, this downside needs to be addressed in order to improve the applicability of the DLA for practical use cases, for instance for online auction-based ticket sales.

As explained by [1], the DLA is explicitly designed as a distribution-free mechanism. This seems a particularly useful property if no knowledge is available about the incoming bidders. If the allocation is repeated regularly, however, it might be possible to infer estimators for distribution parameters from historical data. In this case, one should define some initial threshold prices based on the given information about the stationary input processes in order to mitigate the problems associated with a big ε. Therefore, a deterministic linear program (LP), in particular replacing stochastic variables with their expected values, might be utilized for estimating meaningful thresholds. Dual prices can be accessed after solving the LP with expected willingness-to-pay and item requests. These shadow prices should be good references points for initial thresholds as they mirror expected values. Because all bidders are homogeneous in expectation, the LP can be solved by accepting all requests as long as no capacity restrictions are violated, just like the *Greedy Algorithm*. Since two resources will still be available when the third one is exhausted due to our assumption of stepwise demands, only one dual price will take a positive value in our setting.

If the allocation problem has to be solved repeatedly with similar input features, it might also be reasonable to use an average over past dual prices as initial thresholds. In particular, when input distributions are known, several simulations can be executed prior to the actual allocation task. For each simulation run, the first set of dual prices after εn bidders is stored. The average over this set of shadow prices, in our case over 10 simulations, can serve as initial thresholds for the actual allocation.

Alternatively, more naïve initial threshold prices are the expected values of the known input distributions for the willingness-to-pay for each resource. All bids, exceeding these prices will be accepted as long as the problem remains feasible.

Table 6. OPT-Approximations with Initial Thresholds

ε	*Baseline Treatment*	*LP Thresholds*	*Simulated Thresholds*	*Expected Values of Distributions*
0.001	94.54%	94.41%	94.74%	94.44%
0.01	93.67%	94.23%	94.84%	94.53%
0.03	90.06%	92.86%	91.66%	93.49%
0.05	86.32%	91.69%	91.10%	92.31%
0.1	75.88%	87.33%	85.75%	89.44%
0.15	65.16%	82.63%	77.95%	86.10%
0.25	47.25%	74.93%	63.51%	80.85%

The results of these different approaches for different ε are presented in Table 6. While the conventional DLA begins to rapidly deteriorate for $\varepsilon \geq 0.05$, initial thresholds keep the performance on a higher level. In particular, using thresholds does not seem to be disadvantageous for any ε. Simulated dual prices appear to be too instance-specific, making it difficult to rely on a limited set of past thresholds for future allocations. Retrieving initial thresholds from solving a LP produces better approximations of the OPT. This especially applies to scenarios where large fractions ε are chosen. Simply employing expected values of the distributions, however, further improves the approximation capabilities and should even be preferred to the more sophisticated alternatives. Again, it should be emphasized that this extension does not come along with any essential drawbacks. For small ε only few bidders are affected, whereas initial thresholds enable major gains for large fractions.

The DLA together with well-defined initial thresholds can therefore be deemed appropriate for complex allocation tasks. It represents a viable alternative to online ticket sales or other multi-item B2C businesses, achieving superior approximations of the OPT while maintaining satisfactory runtimes at the same time.

Result 6. *Defining initial threshold prices by using available information on the input distributions mitigates the drawbacks of the DLA associated with very large learning fractions ε. Matching the thresholds with the expected willingness-to-pay until the first calibration of dual prices already aids in alleviating revenue loss. It can also improve on the perceived fairness in handling the bidders and thus enhances the practical applicability of the DLA substantially.*

4 Conclusion

Online auctions represent a promising alternative to conventional posted price mechanisms, potentially enabling a more effective exploitation of customer willingness-to-pay. Based on the notion of ticket sales, we conducted an experimental study on two seminal algorithms proposed by [1] and put special emphasis on the practicability of their underlying primal-dual framework. Both the OLA and the DLA were implemented for the purpose of simulation-based experimental testing, along with some intuitive benchmarks, ranging from a quick-and-easy *Greedy Algorithm* to a computationally intensive *Interval Learner*.

There is a fundamental trade-off between the capability to approximate the ex-post optimal revenue and the average computational runtime. We ran extensive numerical experiments to discover dependencies and sensitivities of these opposing objectives. Through precisely defined re-calibrations of dual threshold prices, the DLA is able to approximate the OPT very well against a stationary process of bids. At the same time, it maintains reasonable runtimes, since dual updates occur more frequently at early stages of the allocation process. Our experiments illustrate that the DLA reacts robust to changes in many input parameters and proves to be extraordinarily dominant in situations of resource scarcity and excess demand. In this case, the decision rules implied by the DLA enable deliberate allocations through accurately determined thresholds.

Addressing the problem of generally rejecting bids in the first learning phase, we drafted several extensions, making use of known distribution information and aiming at the definition of some initial thresholds that could be employed until the first ordinary calibration. While retrieving a set of dual prices from a deterministic linear model seems an elegant solution, simply using the expected values of the known input distributions as initial thresholds proves to be an easy and well-performing alternative. Initial threshold prices permit major enhancements with respect to the real-world applicability of the DLA. Together with its robustness regarding other treatment variables, the DLA can thus be viewed as an expedient alternative when it comes to online ticket sales or other revenue-maximizing multi-item allocation tasks.

The basic idea underlying this research paper can be extended in various ways. An empirical validation with real-world data would, for instance, be desirable. Especially the context of online ticket sales seems to be a suitable use case. Furthermore, the algorithms have only been benchmarked with rather straightforward mechanisms. A more extensive experimental study with alternative sophisticated algorithms might provide useful insights into the applicability of different online auction mechanisms. Including risk considerations or non-monetary objectives would only be two possible extensions referring to the DLA.

References

1. Agrawal, S., Wang, Z., Ye, Y.: A Dynamic Near-Optimal Algorithm for Online Linear Programming. Operations Research 62 (4), 876–890 (2014)
2. Babaioff, M., Immorlica, N., Kempe, D., Kleinberg, R.: Online Auctions and Generalized Secretary Problems. ACM SIGecom Exchanges 7 (2), 1–11 (2008)
3. Buchbinder N., Jain K., Naor J.: Online Primal-Dual Algorithms for Maximizing Ad-Auctions Revenue. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) Algorithms – ESA 2007. LNCS, vol. 4698, pp. 253–264. Springer, Berlin, Heidelberg (2007)
4. Buchbinder N., Naor J.: Improved Bounds for Online Routing and Packing via a Primal-Dual Approach. In: 2006 47th Annual IEEE Symposium on Foundations of Computer Science, pp. 293–304. FOCS '06, Berkeley (2006)
5. Buchbinder N., Naor J.: Online Primal-Dual Algorithms for Covering and Packing. Mathematics of Operations Research 34 (2), 270–286 (2009)
6. Cai, Y., Devanur, N.R., Weinberg, S.M.: A Duality Based Unified Approach to Bayesian Mechanism Design. ACM SIGecom Exchanges 15 (1), 71–77 (2016)
7. Devanur, N.R.: Online Algorithms with Stochastic Input. ACM SIGecom Exchanges 10 (2), 40–49 (2011)
8. Devanur, N.R., Hayes T.P.: The Adwords Problem: Online Keyword Matching with Budgeted Bidders under Random Permutations. In: Proceedings of the 2009 ACM Conference Electronic Commerce, pp. 71–78. EC '09, Stanford (2009)
9. Dütting, P., Feldman, M., Kesselheim, T., Lucier, B.: Prophet Inequalities Made Easy: Stochastic Optimization by Pricing Non-Stochastic Inputs. In: 2017 IEEE 58th Annual Symposium on Foundations of Computer Science, pp. 540–551. FOCS '17, Berkeley (2017)
10. Feldman, J., Henzinger, M., Korula, N., Mirrokni, V., Stein, C.: Online Stochastic Packing Applied to Display Ad Allocation. In: de Berg, M., Meyer, U. (eds.) ESA 2010, Part I. LNCS, vol. 6346, pp. 182–194. Springer, Berlin, Heidelberg (2010)
11. Feldman, J., Korula, N., Mirrokni, V., Muthukrishnan, S., Pál, M..: Online Ad Assignment with Free Disposal. In: Leonardi, S. (ed.) WINE 2009. LNCS, vol. 5929, pp. 374–385. Springer, Berlin, Heidelberg (2009)
12. Gönsch, J., Klein, R., Neugebauer, M., Steinhardt, C.: Dynamic Pricing with Strategic Customers. Journal of Business Economics 83 (5), 505–549 (2013)
13. Hajiaghayi, M.T., Kleinberg, R., Sandholm, T.: Automated Online Mechanism Design and Prophet Inequalities. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, pp. 58–65. AAAI '07, Vancouver (2007)
14. Kleinberg, R.: A Multiple-Choice Secretary Algorithm with Applications to Online Auctions. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 630–631. SODA '05, Vancouver (2005)
15. Kriegel, H.P., Schubert, E., Zimek, A.: The (Black) Art of Runtime Evaluation: Are We Comparing Algorithms or Implementations? Knowl Inf Syst 52 (2), 341–378 (2017)
16. Strauss, A.K., Klein, R., Steinhardt, C.: A Review of Choice-Based Revenue Management: Theory and Methods. European Journal of Operational Research 271 (2), 375–387 (2018)
17. Zhong, L., Huang, Q., Wu, F., Chen, G.: TRADE: A Truthful Online Combinatorial Auction for Spectrum Allocation in Cognitive Radio Networks. Wireless Communications and Mobile Computing 15 (9), 1320–1330 (2015)