Randomized algorithms for online bounded bidding

Leah Epstein* Asaf Levin†

Abstract

In the online bidding problem, a bidder is trying to guess a positive number T, by placing bids until the value of the bid is at least T. The bidder is charged with the sum of the bids. In the *bounded* online bidding problem, a parameter k is given, and the bidder is charged only with the largest k bids. It is known that the online bidding problem admits a 4-competitive deterministic algorithm, and an e-competitive randomized algorithm, and these results are best possible. The deterministic best possible competitive ratio for the online bounded bidding problem is also known, for any value of k.

We study the randomized bounded online bidding problem, and show that for any $k > 2$, randomization is helpful, that is, it allows to design an algorithm of a smaller competitive ratio compared to the best deterministic algorithm. In contrast, for $k = 2$, we show a lower bound of 2 on the competitive ratio of any randomized algorithms, matching the upper bound achieved by a trivial deterministic algorithm, which tests all possible bids sequentially.

1 Introduction

The ONLINE BIDDING PROBLEM (OB) is defined as follows. An algorithm needs to submit *bids*, $b_1, b_2, \ldots$, where $b_i > 0$, until for some $q \geq 1$, its bid, b_q, is at least as large as an unknown threshold $T \in \mathbb{N}$. The cost of the algorithm, denoted by $\Delta(T)$, is the sum of its bids, i.e., $\sum_{i=1}^{q} b_i$. The problem is considered to be an online problem, and it is analyzed via competitive analysis. Since an optimal offline algorithm which knows the value T can simply place a single bid of T, an algorithm is $\mathcal{C}$-competitive if for any threshold $T \in \mathbb{N}$, $\Delta(T) \leq \mathcal{C} \cdot T$. While the name *online bidding* was coined only recently [3], in that paper, OB is referred to as a "folklore" online problem. Problems related to OB were frequently studied (see the recent survey article of Chrobak and Kenyon-Mathieu [4] and references therein). In the survey [4], a number of related problems is discussed. Among those problems, there are problems whose study requires reduction to OBB. Two examples of such problems are *incremental medians* [3, 5], and *load balancing on uniformly related machines* [1].

Chrobak et al. [3] stated the following results as folklore: the best possible deterministic competitive ratio for OB is 4, and it is achieved by placing bids for all powers of 2. A folklore result regarding randomized algorithms for OB is an e-competitive algorithm, where $e \approx 2.718$ (see also [2]). Chrobak et al. [3] showed that this bound is best possible.

We note that this problem is equivalent to the following variant. An algorithm needs to submit a *bid set* $\mathbb{B}$ which is a set of positive real numbers. The cost of the algorithm for a bid value $T \in \mathbb{R}^+$ is $\Delta(T) =$

*Department of Mathematics, University of Haifa, 31905 Haifa, Israel. `lea@math.haifa.ac.il`.

†Chaya fellow. Faculty of Industrial Engineering and Management, The Technion, 32000 Haifa, Israel. `levinas@ie.technion.ac.il`.

$\sum_{t \in \mathbb{B}: t<T} t + \inf_{t \in \mathbb{B}: t \geq T} t$. Then an algorithm is $\mathcal{C}$-competitive if for any threshold $T \in \mathbb{R}^+$, $\Delta(T) \leq \mathcal{C} \cdot T$. Clearly the two problems are equivalent, that is, the best competitive ratio (either randomized or deterministic) is the same for the two variants. In this paper to simplify the notations we use the second formulation of the problem.

The ONLINE BOUNDED BIDDING PROBLEM (OBB), with the positive integer parameter $k \geq 2$, is defined as follows. An algorithm again submits a bid set $\mathbb{B}$, but it is charged only with its k largest bids. That is, $\Delta_k(T)$ is defined as the sum of following two values: the first value is the infimum element of $\mathbb{B}$ which is at least T, and the second value is the supremum sum of any $k-1$ elements of $\mathbb{B}$ which are smaller than T. The algorithm is said to be $\mathcal{C}$-competitive if for all $T \in \mathbb{R}^+$, we have $\Delta_k(T) \leq \mathcal{C} \cdot T$. In this paper, we consider randomized algorithms. A randomized algorithm uses a distribution over sets $\mathbb{B} \subseteq \mathbb{R}^+$. The cost of a randomized algorithm is the expected value of $\Delta_k(T)$. Therefore, a randomized algorithm is $\mathcal{C}$-competitive if for all $T \in \mathbb{R}^+$, we have $\mathbf{E}(\Delta_k(T)) \leq \mathcal{C} \cdot T$, where $\mathbf{E}(\cdot)$ is the expected value of its argument.

Paulus, Ye and Zhang [6] have studied a related online batch scheduling problem with batches of fixed capacity, but their results are valid for OBB. They showed that the best possible deterministic competitive ratio for the parameter k is

$$\min_{x \geq 1} \left\{ \frac{x^{k-1} + x^{k-2} + \ldots + x + 1}{x^{k-2}} \right\}.$$

We consider the randomized bounded online bidding problem. For $k = 2$, and $k = 3$, the deterministic algorithms implied by the results of [6] simply let $\mathbb{B} = \mathbb{R}^+$. We show that for $k = 2$ this algorithm is still a best possible randomized algorithm. In contrast, for any $k \geq 3$, we design an improved randomized algorithm. The competitive ratio of our improved randomized algorithm tends to e as k approaches infinity. We note that the competitive ratio converges very fast to e as k grows. This can be observed by the line marked “Randomized UB” in the following table, in which we compare the tight values for deterministic algorithms proven by [6, 3], and the randomized upper bound shown in this paper.

k	2	3	4	5	6	7	8	∞
Deterministic bounds	2 [6]	3 [6]	3.61 [6]	3.83 [6]	3.92 [6]	3.96 [6]	3.98 [6]	4 [3]
Randomized UB	2 [6]	2.5243	2.6582	2.6981	2.7112	2.7157	2.71736	e [3]
Value of z_k	1 [6]	2.047	2.4556	2.6122	2.6747	2.7003	2.71085	e [3]

Table 1. Tight values for the competitive ratio of deterministic algorithms versus an upper bound on the randomized competitive ratio.

2 A randomized lower bound for $k = 2$

In this section we construct a lower bound of 2 in the competitive ratio of any online algorithm for OBB. Our lower bound is inspired by the randomized lower bound of e for OB due to [3].

Theorem 1 *The competitive ratio of any randomized online algorithm for* OBB *with* $k = 2$ *is at least 2.*

Proof. We fix a randomized online algorithm $\mathcal{A}$ whose competitive ratio is β. Let $n \in \mathbb{N}$ be a sufficiently large number. We will consider an adversary which uses only threshold values of T which are integers between 1 and n. Since our adversary uses only such values, we can modify algorithm $\mathcal{A}$ to use only integer numbers between 1 and n (such that the probability that n is included in the bid set, which is selected by

$\mathcal{A}$, is 1) without increasing the competitive ratio of the algorithm against our adversary. This modification is carried out by replacing every member of the bid set selected by $\mathcal{A}$ with its floor value. In this way, for every threshold value, the cost paid by the algorithm may only decrease and thus the algorithm remains β-competitive. Therefore, in the remainder of this proof, we restrict ourselves to an adversary which uses threshold values of T which are integers between 1 and n, and consider a randomized algorithm $\mathcal{A}$ with support containing bid sets which are subsets of $\{1, 2, \ldots, n\}$. We let $[n] = \{1, 2, \ldots, n\}$.

We define the following probabilities for each pair $a, b \in [n]$, where $a < b$. $X(a, b)$ is the probability that a and b are two consecutive values in the bid set $\mathbb{B}$ selected by $\mathcal{A}$. Then, the algorithm pays (in expectation) $\sum_{a,b\in[n]:a<T\leq b} (a + b) \cdot X(a, b)$ against threshold T. Since it is a β-competitive algorithm, this value is at most βT. We also have $\sum_{a,b\in[n]:a<T\leq b} X(a, b) \geq 1$, which holds since for every threshold value $T \in [n]$ and for every realization of the bid set $\mathbb{B}$, we must have two consecutive values in $\mathbb{B}$ for which T is between them (i.e., $a < T$ and $T \leq b$).

Hence, β and the vector X must form a feasible solution to the following linear program.

$$\begin{array}{llll}
\min & & \beta & \\
s.t. & & \displaystyle\sum_{a,b\in[n]:a<T\leq b} X(a,b) \geq 1 & \forall T \in [n] \\
& \beta - & \displaystyle\sum_{a,b\in[n]:a<T\leq b} \frac{(a+b)}{T} \cdot X(a,b) \geq 0 & \forall T \in [n] \\
& & X(a,b) \geq 0 & \forall a, b \in [n], a < b.
\end{array}$$

Clearly, the competitive ratio of $\mathcal{A}$ is at least the value β^* of the optimal solution of the last linear program. To get a lower bound on β^*, we will present a feasible solution to its dual linear program which is presented next. The dual linear program can be stated as follows, where $(\mu(T))_{T\in[n]}$ are the dual variables corresponding to the first set of constraints, and $(\pi(T))_{T\in[n]}$ are the dual variables corresponding to the second set of constraints.

$$\begin{array}{llll}
\max & & \displaystyle\sum_{T=1}^{n} \mu(T) & \\
s.t. & & \displaystyle\sum_{T=1}^{n} \pi(T) \leq 1 & \\
& \displaystyle\sum_{T=a+1}^{b} \mu(T) - & \displaystyle\sum_{T=a+1}^{b} \frac{a+b}{T}\pi(T) \leq 0 & \forall a, b \in [n], a < b \\
& & \mu(T), \pi(T) \geq 0 & \forall T \in [n].
\end{array}$$

We would like to remove the first constraint. To do so, we note that given two vectors $(\mu(T))_{T\in[n]}$ and $(\pi(T))_{T\in[n]}$ which satisfy the other constraints, and are not identically equal to zero, we can scale the vectors by a multiplicative factor of $\sum_{T=1}^{n} \pi(T)$ to create a solution for the dual linear program. (Note that if $\sum_{T=1}^{n} \pi(T) = 0$, then $\mu(T) = \pi(T) = 0$ holds for any $T \in [n]$.) As a result of this scaling, the objective function value is also scaled by a factor of $\sum_{T=1}^{n} \pi(T)$. Hence our dual linear program is equivalent to the following mathematical program.

$$\begin{array}{lll} \max & \frac{\sum_{T=1}^{n} \mu(T)}{\sum_{T=1}^{n} \pi(T)} & \\ s.t. & \sum_{T=a+1}^{b} \mu(T) - \sum_{T=a+1}^{b} \frac{a+b}{T}\pi(T) \leq 0 & \forall a,b \in [n], a < b \\ & \mu(T), \pi(T) \geq 0 & \forall T \in [n]. \end{array}$$

We next consider the vectors $\mu(T) = 2T - 1$ and $\pi(T) = T$ for all $T \in [n]$. We first note that the objective function value is $\frac{\sum_{T=1}^{n} \mu(T)}{\sum_{T=1}^{n} \pi(T)} = \frac{\sum_{T=1}^{n} (2T-1)}{\sum_{T=1}^{n} T} = 2 - \frac{n}{n \cdot \frac{n+1}{2}} = 2 - \frac{2}{n+1}$ which tends to 2 as n approaches infinity.

It remains to show that this is a feasible solution for every value of n. First note that $\mu(T), \pi(T) \geq 0$ for all $T \in [n]$, and hence it remains to consider the set of constraints $\sum_{T=a+1}^{b} \mu(T) - \sum_{T=a+1}^{b} \frac{a+b}{T}\pi(T) \leq 0$. We fix $a, b \in [n]$ such that $a < b$, and consider the corresponding constraint.

$$\begin{aligned} & \sum_{T=a+1}^{b} \mu(T) - \sum_{T=a+1}^{b} \frac{a+b}{T}\pi(T) \\ = \ & \sum_{T=a+1}^{b} (2T-1) - \sum_{T=a+1}^{b} \frac{a+b}{T} T \\ = \ & 2 \sum_{T=a+1}^{b} T - (b-a) - (a+b)(b-a) \\ = \ & 2 \cdot \left(\frac{a+1+b}{2} \right) \cdot (b-a) - (b-a) - (a+b)(b-a) \\ = \ & 0 \ , \end{aligned}$$

and the claim follows. ■

3 An improved randomized algorithm for $k \geq 3$

The algorithm in this section is an adaptation of the e-competitive algorithm for OB, and similarly to the algorithm of [6] uses doubling with a parameter which is a function of k.

We denote by $z_k > 1$ a constant value which will be determined later. Our algorithm picks uniformly at random a value u with uniform distribution in the interval $[0, 1)$, that is $u \sim U[0, 1]$. The resulting bid set is defined as $\mathbb{B}_u = \{z_k^{i+u} : i \in \mathbb{Z}\}$. It remains to analyze the competitive ratio of this algorithm.

Given a threshold value T, we denote by $T_u = \min_{i \in \mathbb{Z}: z_k^{i+u} \geq T} z_k^{i+u}$, that is, the smallest value in the bid set $\mathbb{B}_u$ which is at least T.

Lemma 2 *For every realization of u, the algorithm pays at most $\frac{1-\frac{1}{z_k^k}}{1-\frac{1}{z_k}} \cdot T_u$.*

Proof. Note that given T and a fixed value of u, the algorithm pays $T_u + \frac{T_u}{z_k} + \frac{T_u}{z_k^2} + \ldots + \frac{T_u}{z_k^{k-1}}$ since the members of $\mathbb{B}_u$ are geometric sequence with the common ratio z_k. The claim follows since

$$T_u + \frac{T_u}{z_k} + \frac{T_u}{z_k^2} + \ldots + \frac{T_u}{z_k^{k-1}} = T_u \cdot \sum_{i=0}^{k-1} \left(\frac{1}{z_k}\right)^i = \frac{1-\frac{1}{z_k^k}}{1-\frac{1}{z_k}} \cdot T_u \ .$$

■

The next corollary is an immediate consequence of the last lemma.

Corollary 3 *The expected value of the cost of* $\mathbb{B}$ *is at most* $\frac{1-\frac{1}{z_k^k}}{1-\frac{1}{z_k}} \cdot \mathbf{E}(T_u)$.

The next step is to bound $\mathbf{E}(T_u)$ in terms of T.

Lemma 4 $\mathbf{E}(T_u) = \frac{z_k-1}{\ln z_k} \cdot T$.

Proof. We consider the random variable $A = \frac{T_u}{T}$. Then, the distribution of A is z_k^{α} where $\alpha \sim U[0,1]$. We next compute the expected value of A. $\mathbf{E}(A) = \int_0^1 z_k^{\alpha} d\alpha = \frac{z_k^{\alpha}}{\ln z_k}|_{\alpha=0}^1 = \frac{z_k-1}{\ln z_k}$, and the claim follows. ■

We conclude that the expected cost of the algorithm is at most

$$\rho(k) = \frac{1-\frac{1}{z_k^k}}{1-\frac{1}{z_k}} \cdot \frac{z_k-1}{\ln z_k} \cdot T \ .$$

Therefore, for every value of $k \geq 3$, we choose z_k as the minimizer of the function $\rho(k)$ and then $\rho(k)$ is the resulting competitive ratio. We note that $\rho(k) = \frac{\frac{z_k^k-1}{z_k^k}}{\frac{z_k-1}{z_k}} \cdot \frac{z_k-1}{\ln z_k} = \frac{z_k^k-1}{z_k^{k-1}\ln z_k}$. From this last formula, it is clear that $\rho(k) < e$ for every finite value of k, since by setting $z_k = e$, we get an expression which is smaller than e and it is clearly an upper bound on $\rho(k)$. z_k is the unique root larger than 1 of the following equation: $(z_k^k + k - 1) \cdot \ln z_k = z_k^k - 1$. Some values of the optimal z_k value and the resulting $\rho(k)$ are presented in Table 1. The following theorem is established.

Theorem 5 *For* $k \geq 3$ *there is a randomized algorithm for* OBB *whose competitive ratio is* $\rho(k) = \frac{z_k^k-1}{z_k^{k-1}\ln z_k}$ *where* z_k *is the unique root larger than 1 of the equation* $(z_k^k + k - 1) \cdot \ln z_k = z_k^k - 1$.

4 Concluding remarks

We have presented improved randomized algorithms for OBB for any $k > 2$, and showed that for $k = 2$, the deterministic algorithm of [6] is the best possible randomized algorithm. We conjecture that our algorithms for $k > 2$ are the best possible randomized algorithms. The current methods for obtaining randomized lower bounds for OB do not extend to OBB. In a nutshell, the difficulty arises since in the analysis of the performance of the online algorithm it is necessary to take into account sets of k consecutive values used by the algorithm's bid set. For $k = 2$ this was possible since the last two values tested for a given threshold are the first one that is larger or equal to the threshold, and its predecessor in the bid set, so these are exactly the two values in the bid set that the threshold lies between them. On the other hand, for $k \geq 3$, additional values of the bid set (but not the entire bid set) needs to be considered. We leave this open problem for future research.

References

[1] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line load balancing with applications to machine scheduling and virtual circuit routing. *J. ACM*, 44:486–504, 1997.

[2] Y. Azar. On-line load balancing. In Amos Fiat and Gergard J. Woeginger, editors, *Online Algorithms: The State of the Art*, pages 178–195. Springer, 1998.

[3] M. Chrobak, C. Kenyon, J. Noga, and N. E. Young. Incremental medians via online bidding. *Algorithmica*, 50(4):455–478, 2008.

[4] M. Chrobak and C. Kenyon-Mathieu. SIGACT news, Online algorithms column 10: Competitiveness via doubling. *SIGACT News*, 37(4):115–126, 2006.

[5] G. Lin, C. Nagarajan, R. Rajamaran, and D. Williamson. A general approach for incremental approximation and hierarchical clustering. In *Proc. of the 17th Symp. on Discrete Algorithms (SODA2006)*, pages 1147–1156, 2006.

[6] J. J. Paulus, D. Ye, and G. Zhang. Optimal online-list batch scheduling. *Information Processing Letters*, 109(19): 1125–1128, 2009.