Supplementary material for "An Online Learning Approach based Trading Strategy for FTR Auction Market"

Guibin Chen, *Student Member, IEEE*, Ye Guo, S*enior Member, IEEE,* Wenjun Tang, Qinglai Guo, *Senior Member, IEEE*, Hongbin Sun, *Fellow, IEEE* and Wenqi Huang

APPENDIX A

The implementation procedure of the TBS algorithm is presented in Alg.1. The procedure includes four main steps: 1) re-updating (line 3-6); 2) exploring solution and output it (line 7-17); 3) updating (line 18-21); 4) expansion (line 22-25).

1) Re-updating: At each round n, the algorithm receives the newly observed information and calculates $\hat{n}$ to decide whether to re-update the performance criterions of all nodes. Such a re-update phase is critical to avoid trapping in local optimum (as the 'current path' in Fig. 3) since other nodes outside current optimistic path may have better performance than current optimistic path P_n . A frequent implementation of the re-updating results in increasing computational complexity. Therefore, $\hat{n} = 2^{\lfloor \log_2 n \rfloor}$ is set as a variable depends on n to control the frequency of re-updating implementation.

Algorithm 1: Tree-based Bid Searching Algorithm

1: **Initialization:** Total rounds: N, $n = 1$, $\mathcal{T}_1 = \{(0,1)\}$, $m, \varepsilon \in (0,1)$, $L, c > 0$.
2: **while** $n \le N$:
3: receive cleared results $(\sum_{h=1}^{H} \Delta\lambda_{h,n-1}, \pi_{n-1})$
4: **if** $n = \hat{n}$:
5: Re-calculate $S_{d,i}$ for all $(d,i) \in \mathcal{T}_n$;
6: Re-calculate $G_{d,i}$ for all $(d,i) \in \mathcal{T}_n$;
#Update Parameter
7: $L = Parameter\ update$
#Formulate optimal path
8: $P_n = \{(0,1)\}$
9: **for** (d,i) in range $((0,1), (d_n, i_n))$:
10: **if** (d,i) is not *Leaf* node and $T_{d,i}(n) \ge T_d^*$:
11: **if** $G_{d+1,2i-1}(n) \ge G_{d+1,2i}(n)$:
12: $(d,i) = (d+1, 2i-1)$;
13: **else:**
14: $(d,i) = (d+1, 2i)$;
15: $P_n = P_n \cup (d,i)$;
16: $(d_n, i_n) = (d,i)$;
17: select bid b_{d_n,i_n} from (d_n, i_n) randomly;
18: observe the payoff gained by b_{d_n,i_n} in auction;
19: calculation $\bar{p}_{d_n,i_n}(n)$ and $T_{d_n,i_n}(n)$;
20: calculate the *Slack upper bound* for (d_n, i_n) ;
21: update the *Tight upper bound* value for $(d,i) \in P_n$;
22: calculate the threshold value as

$$T_{d_n}^* = \frac{c^2}{L^2 (m)^{2d_n}} \ln\left(\frac{\sqrt{4N - \hat{n}}}{\varepsilon}\right)$$

#Partitoion node covering space
23: **if** $T_{d_n,i_n}(n) \ge T_{d_n}^*$ and (d_n, i_n) is *Leaf* node:
24: $(d_n, i_n) = (d_n + 1, 2i_n - 1) \cup (d_n + 1, 2i_n)$
25: $\mathcal{T}_n = \mathcal{T}_n \cup \{(d_n + 1, 2i_n - 1) \cup (d_n + 1, 2i_n)\}$
26: $S_{d_n+1,2i_n-1} = S_{d_n+1,2i_n} = \infty$
27: $n = n + 1$
28: **end**

Algorithm 2: Parameter update

1: **Input:** $\mathcal{T}_n$, m, L.
2: **for** (d,i) in range $((0,1), (d_n, i_n))$:
3: $gap = \left|\bar{p}_{d,i}^{lower\ bound} - \bar{p}_{d,i}^{upper\ bound}\right|$
4: **if** $gap > Lm^d$:
5: $L = gap/m^d + \delta$
6: **Output**: L

2) *Exploring solution and output it*: After the re-updating phase, the algorithm starts to formulate an optimistic path from the root node $(0, 1)$ until find the node with relatively higher resolution of covering space, TBS formulates the path under the guidance of criterions of node (d,i). To satisfy *Assumption 2* in practice, the value of L is updated by newly collected practical data. Alg. 2 presents details of updating L value in which the δ is a small positive constant to adaptively adjust the result.

One of these criterions for node (d,i) is the empirical mean-profit $\bar{p}_{d,i}(n)$ associated with bidding price $b_{d,i}$ from node (d,i), which is calculated as

$$\bar{p}_{d,i}(n) = \frac{1}{n-1} \sum_{n'=1}^{n-1} \left(\sum_{h=1}^{H} \Delta\lambda_{n',h} - \pi_{n'} \right) \mathbb{1}\{b_{d,i} \ge \pi_{n'}\} \quad (9)$$

As shown in (9), criterions are calculated based on newest available information in each round n, which not only facilitates the algorithm to acquire more reliable estimation of underlying joint distribution of $(\sum_{h=1}^{H} \Delta\lambda_h, \pi)$, but also makes the algorithm to adaptively update its policy.

Although a node (d,i) is corelated to a bidding price $b_{d,i}$, it also covers a proportion of bidding space $\mathcal{B}$, namely the $\mathcal{R}_{d,i}$, which makes the mean-profit $\bar{p}_{d,i}(n)$ not an accurate criterion to describe node (d,i). Based on the mean profit corresponding to node (d,i), two criterions are further defined in an optimistic fashion to upper bound potential profit that can be achieved by node (d,i). TBS defines the first criterion $S_{d,i}(n)$ as Slack upper bound. For each node $(d,i)\in\mathcal{T}_n$, its Slack upper bound $S_{d,i}(n)$is calculated as

$$S_{d,i}(n)=\begin{cases}\bar{p}_{d,i}(n)+L(m)^d+c\sqrt{\dfrac{\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)}{T_{d,i}(n)}}, T_{d,i}(n)>0;\\ \infty,\quad otherwise.\end{cases} \tag{10}$$

where the first term represents the mean payoff acquired by node (d,i) up to round n. The second term denotes the uncertainties of node's performance caused by the size of node's covering bidding space, $\mathcal{R}_{d,i}$. $T_{d,i}(n)$ in the third term represents the times that node (d,i) has been selected up to round n. Thus, we use the third term to describe the uncertainties caused by the insufficient selecting times of node (d,i).

As the TBS uses the binary tree $\mathcal{T}_n$, the upper bound of node (d,i)'s potential profit is also reflected by its descendants. To guarantee the validity of upper bound of node (d,i) with enough probability, TBS defines a *Tight upper bound* $G_{d,i}(n)$ for node (d,i)

$$G_{d,i}(n)=\min\left[S_{d,i}(n),\max\left(G_{d+1,2i-1}(n),G_{d+1,2i}(n)\right)\right] \tag{11}$$

As shown in (11), $G_{d,i}(n)$ aims to approximate a more accurate, or higher probability upper bound of for node (d,i). The calculation of *Tight upper bound* of nodes in the tree is destined to be a backward process. For nodes that have no child node, the *Tight upper bound* exactly equals to the *Slack upper bound*: $G_{d,i}(n)=S_{d,i}(n)$.

TBS formulates the optimistic path P_n from tree $\mathcal{T}_n$'s root node by selecting nodes with the highest $G_{d,i}(n)$ until it reaches the deepest node (d_n,i_n) in the path, which is either a leaf node or an internal node which has not been explored sufficiently, i.e., $T_{d_n,i_n}(n)<T^*_{d_n}$. Then the bidding price $b_{d_n,i_n}\in\mathcal{R}_{d_n,i_n}$ is outputted by the algorithm as the optimal bidding price up to round *n*.

3) *Updating*: After outputting the bidding price, TBS conducts an update estimate for all nodes in the path P_n. Since the calculation for $G_{d,i}(n)$ for all $(d,i)\in P_n$ requires to be conducted backwardly, $\bar{p}_{d_n,i_n}(n)$ and $G_{d_n,i_n}(n)$ are primarily calculated for the deepest node (d_n,i_n). As the deepest node of current formulated path P_n is (d_n,i_n), which may be the node that contain optimal bidding price. To identify whether to expand node (d_n,i_n), its threshold $T^*_{d_n}$is calculated in this step. The choice of $T^*_{d_n}$ will be discussed in next step.

4) *Expansion*: The expansion is the key step for the algorithm to find the solution, namely the optimal bidding price. As shown in Eq. (10), the upper bound $S_{d,i}(n)$ for node (d,i) consists of the empirical mean profit $\bar{p}_{d,i}(n)$ and two additional term, whereas the first term represents the largest variations of empirical mean profit gained by two bidding prices from $\mathcal{R}_{d,i}$, and the second term denotes uncertainties on estimation which decreases with the number of selection this node. Specially, when the third term is smaller than the second term, it is the uncertainties caused by node's size that dominates. To increase the resolution of current approximation of optimal bidding price, the covering bidding space $\mathcal{R}_{d,i}$ requires to be partitioned. We expand the node and add its descendants to tree $\mathcal{T}_n$ when it has been selected sufficiently. These occur when $T_{d,i}(n)\geq T^*_d$, where

$$L(m)^d=c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)}{T^*_d(n)}} \tag{12}$$

$$\Rightarrow T^*_d=\frac{c^2}{L^2(m)^{2d}}\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right) \tag{13}$$

As shown in Eq. (13), T^*_d decreases with the number of rounds. The design drives algorithm to spend more time on exploring covering space $\mathcal{R}_{d,i}$ with large size at the initial stage, whereas the threshold value decreases because of a larger $\hat{n}$, which accelerates process of increasing resolution of node in later steps. At the point that $T_{d,i}(n)\geq T^*_d$, TBS sets the *S*-value of these two children nodes to ∞, which enforces the algorithm to select them in next steps.

APPENDIX B

First, we would like to present the identification of select times' threshold value T^*_d in detail. Recall the *Slack upper bound* function for nodes on the tree, once the last two additional terms are equivalent, the original node covering bidding space need to be partitioned. The mathematical formulation is

$$Lm^d=c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)}{T^*_d(n)}}\Rightarrow T^*_d=\frac{c^2}{L^2m^{2d}}\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right) \tag{27}$$

where $\hat{n}=2^{\lfloor\log_2 n\rfloor}$, denoting the occasion to re-update all nodes' *Slack upper bound* and *Tight upper bound* value. Therefore, we have that $n\leq\hat{n}\leq 2n$, which leads to following inequalities

$$\begin{aligned}\frac{c^2}{L^2}m^{-2d}\leq\frac{c^2}{L^2}\ln\left(\frac{\sqrt{4N-2n}}{\varepsilon}\right)m^{-2d}\leq T^*_d\\ \leq\frac{c^2}{L^2}\ln\left(\frac{\sqrt{4N-n}}{\varepsilon}\right)m^{-2d}\end{aligned} \tag{28}$$

Before the demonstration on the growth rate of tree's depth, we'll primarily propose the concept of nodes' number. We use the notation $\mathcal{T}_n$ to represent the tree's structure up to round n. Meanwhile, $\mathcal{A}_d(n)$ denotes the number of nodes on depth d of $\mathcal{T}_n$ and $\mathcal{A}^+_d(n)$ denotes the number of parent nodes at depth d on $\mathcal{T}_n$. In addition, we use $\mathcal{D}(n)$ to denote the depth of $\mathcal{T}_n$ up to round n. Thus, we have

$$n=\sum_{d=0}^{\mathcal{D}(n)}\sum_{i\in\mathcal{A}_d(n)}T_{d,i}(n)\geq\sum_{d=0}^{\mathcal{D}(n)-1}\sum_{i\in\mathcal{A}_d(n)}T_{d,i}(n)$$

$$\geq \sum_{d=0}^{\mathcal{D}(n)-1} \sum_{i\in\mathcal{A}_d^+(n)} T_{d,i}(n) \geq \sum_{d=0}^{\mathcal{D}(n)-1} T_d^* \geq \sum_{d=0}^{\mathcal{D}(n)-1} \frac{c^2}{L^2} m^{-2d}$$
$$\geq \frac{(cm)^2}{L^2} m^{-2\mathcal{D}(n)} \sum_{d=1}^{\mathcal{D}(n)-1} m^{-2(d-\mathcal{D}(n)+1)} \geq \frac{(cm)^2}{L^2} m^{-2\mathcal{D}(n)} \tag{29}$$

From above inequalities, we could obtain that

$$\mathcal{D}(n) \leq \frac{\ln\left(\frac{nL^2}{(cm)^2}\right)}{2(1-m)} \tag{30}$$

APPENDIX C

Since we use the empirical average payoff to approximate the expected payoff, a demonstration required to be presented to guarantee the validity and reliability. Firstly, the event could be described as

$$\{\mathcal{H}_n\} = \left\{ \left|\bar{p}_{d,i}(n) - \mu(b_{d,i})\right| \leq c\sqrt{\frac{\ln\frac{\sqrt{4N-2n}}{\varepsilon}}{T_{d,i}(n)}} \right\} \tag{31}$$

where $(d,i) \in \mathcal{T}_n, n = 1,2,3, \dots N$.

The complementary event of $\mathcal{H}_n$ is denoted by $\mathcal{H}_n^c$. The probability of $\mathcal{H}_n^c$ is bounded as

$$P[\mathcal{H}_n^c] \leq (\sum_{(d,i)\in\mathcal{T}_n} \sum_{T_{d,i}(n)}^{n} P[\left|\bar{p}_{d,i}(n) - \mu(b_{d,i})\right| \geq c\sqrt{\frac{\ln\frac{\sqrt{4N-2n}}{\varepsilon}}{T_{d,i}(n)}})$$
$$\leq_{(1)} \sum_{(d,i)\in\mathcal{T}_n} \sum_{T_{d,i}(n)}^{n} 2\exp\left(-2T_{d,i}(n) * c^2 \frac{\ln\left(\frac{\sqrt{2N-2n}}{\varepsilon}\right)}{T_{d,i}(n)}\right)$$
$$= 2\exp\left(-2c^2 \ln\left(\frac{\sqrt{4N-2n}}{\varepsilon}\right)\right) n \sum_{d=0}^{\mathcal{D}(n)} \mathcal{A}_d(n) \tag{32}$$

The inequality (1) is by Chernoff-Hoeffding inequality. According to Lemma 1, the maximum depth for $\mathcal{T}_n$ is bounded by $\ln\left(\frac{nL^2}{(cm)^2}\right)/2(1-m)$, thus, $\sum_{d=0}^{\mathcal{D}(n)} \mathcal{A}_d(n)$ could be upper bounded as

$$\sum_{d=0}^{\mathcal{D}(n)} \mathcal{A}_d(n) \leq_{(2)} 2^{\left(\frac{\ln\left(\frac{nL^2}{(cm)^2}\right)}{2(1-m)}\right)+1} \leq 2\left(\frac{nL^2}{(cm)^2}\right)^{\frac{1}{2(1-m)}} \tag{33}$$

where inequality (2) is based on the extreme condition that the tree filled all bidding space at each depth, which would not happen since if a parent node demonstrated as non-promising, then the algorithm shall not partition it. Therefore, the upper bound here is appropriate.

Then the upper bound of probability of $\mathcal{H}_n^c$ is as

$$P[\mathcal{H}_n^c] \leq 4n\left(\frac{\varepsilon}{\sqrt{4N-2n}}\right)^{2c^2}\left(\frac{nL^2}{(cm)^2}\right)^{\frac{1}{2(1-m)}} \tag{34}$$

The value of c and ε could be set as

$$c = 3\sqrt{\frac{1}{1-m}}, \qquad \varepsilon = \sqrt[18]{\frac{m}{L}}$$

Then, the upper bound of probability of $\mathcal{H}_n^c$ could be rewrite as

$$P[\mathcal{H}_n^c] \leq 4n\left(\frac{1}{4N-2n}\right)^{\frac{9}{1-m}}\left(\frac{m}{3L}\right)^{\frac{1}{1-m}} n^{\frac{1}{2(1-m)}}\left(\frac{L\sqrt{1-m}}{2m}\right)^{\frac{1}{1-m}}$$
$$= 4n^{1-\frac{9}{1-m}+\frac{1}{2(1-m)}}\left(\frac{1}{2}\right)^{\frac{4}{1-m}}\left(\frac{\sqrt{1-m}}{6}\right)^{\frac{1}{1-m}}$$
$$= \frac{1}{24} n^{\frac{-2m-15}{2(1-m)}} \leq \frac{1}{24} n^{-\frac{15}{2}} < \frac{1}{24} n^{-7} \tag{35}$$

Therefore, the probability of event $\mathcal{H}_n$ is no less than $1 - \frac{1}{24}n^{-7}$.

APPENDIX D

Define the instantaneous regret at round n as $\Delta_n = \mu^* - p_n$, where $p_n = (\Delta\lambda_n - \pi_n)\mathbb{I}\{b_n \geq \pi_n\}$. For total N auction rounds, the regret could be defined as

$$R_N = \sum_{n=1}^{N} \Delta_n = \sum_{n=1}^{N} \Delta_n \mathbb{I}\mathcal{H}_n + \sum_{n=1}^{N} \Delta_n \mathbb{I}\mathcal{H}_n^c$$
$$= R_N^{\mathcal{H}} + R_N^{\mathcal{H}^c} \tag{36}$$

The value of $R_N^{\mathcal{H}^c}$ could be further decomposition as

$$R_N^{\mathcal{H}^c} = \sum_{n=1}^{\sqrt{N}} \Delta_n \mathbb{I}\mathcal{H}_n^c + \sum_{n=\sqrt{N}+1}^{N} \Delta_n \mathbb{I}\,\mathcal{H}_n^c \tag{37}$$

where the first term could be easily bounded by $\sqrt{N}$ since that Δ_n no larger than 1. Then the second term could be bounded as

$$P[\cup_{n=\sqrt{N}+1}^{N} \mathcal{H}_n^c] \leq \sum_{n=\sqrt{N}+1}^{N} P[\mathcal{H}_n^c] \leq \sum_{n=\sqrt{N}+1}^{N} \frac{1}{24} n^{-7}$$
$$\leq \int_{\sqrt{N}}^{+\infty} \frac{1}{24} n^{-7} dn \leq \frac{1}{144N^3} \tag{38}$$

Such result shows that the probability to have the event $\mathcal{H}_n^c$ after $\sqrt{N}$ rounds is less than $\frac{1}{144N^3}$, which could be neglected without loss of generality.

Therefore, the regret under the condition that event $\mathcal{H}_n^c$ holds could be upper bounded as

$$R_N^{\mathcal{H}^c} \leq \sqrt{N} \tag{39}$$

which holds with probability $1 - \frac{1}{144N^3}$.

APPENDIX E

Decompose the instantaneous regret as

$$\Delta_n = \mu^* - p_n = \mu^* - \mu(b_{d_n,i_n}) + \mu(b_{d_n,i_n}) - p_n$$
$$= \Delta_{d_n,i_n} + \hat{\Delta}_n \tag{40}$$

Thus, the regret under condition that event $\mathcal{H}_n$ hold could be further expressed as

$$R_N^{\mathcal{H}} = \sum_{n=1}^{N} \Delta_{d_n,i_n} \mathbb{I}\{\mathcal{H}_n\} + \sum_{n=1}^{N} \hat{\Delta}_n \mathbb{I}\{\mathcal{H}_n\} \tag{41}$$

We first discuss the upper bound for the second term. Based on the Azuma inequality, the second term of $R_N^{\mathcal{H}}$ is bounded as

$$\sum_{n=1}^{N} \hat{\Delta}_n \mathbb{I}\{\mathcal{H}_n\} \leq \sqrt{2N\ln N} \tag{42}$$

which holds with probability $1 - \frac{1}{N}$.

Recall Assumption 2, the differences of expected payoff between bids from the same node covering bidding space are bounded by the space size. Thus, we have

$$\mu^* - \mu\left(b_{d_n,i_n}(n)\right) \leq Lm^{d_n} \tag{43}$$

Noted that (d_n, i_n) denotes the node selected at round n. Meanwhile, for nodes in current optimal path $\mathcal{P}_n$, an inequality chain could be developed based on the definition of nodes' Tight upper bound as

$$\begin{aligned} G_{d,i}(n) &= \min\left[S_{d,i}(n); \max\left(G_{d+1,2i-1}, G_{d+1,2i}\right)\right] \\ &\leq \max\left(G_{d+1,2i-1}, G_{d+1,2i}\right) \leq G_{d+1,i^*} \end{aligned} \tag{44}$$

where i^* denotes the rank of node with maximum G value.

Obviously, such inequality chain could extend from the root node to last node of current optimal path $\mathcal{P}_n$. Then we could derive that $G_{d,i}(n) \leq S_{d,i}(n) \leq G_{d+1,i^*}$.

Since the root node covers the entire bidding space, thus, there exist at least one node containing optimal bid b^* in $\mathcal{P}_n$. Denoted the bidding space that contains optimal bid as (d^*, i^*), for which $d^* < d_n$. The *Slack upper bound* of (d^*, i^*) could be bounded as

$$\begin{aligned} S_{d^*,i^*}(n) &= \bar{p}_{d^*,i^*}(n) + Lm^{d^*} + c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)}{T_{d^*,i^*}(n)}} \\ &\geq \bar{p}_{d^*,i^*}(n) + Lm^{d^*} + c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-2n}}{\varepsilon}\right)}{T_{d^*,i^*}(n)}} \\ &\geq_{(3)} \mu\left(b_{d^*,i^*}(n)\right) + Lm^{d^*} \geq \mu^* \end{aligned} \tag{45}$$

here inequality (3) is developed based on Assumption 2.

Noted that (d^*, i^*) would further generate more descendants, some of which would also contain b^*. Assume a leaf node (d_f, i_f) as a descendant node of (d^*, i^*) which also contains b^*. Therefore, for node (d_f, i_f), we could easily obtain that $S_{d_f,i_f}(n) \geq \mu^*$ and $S_{d_f,i_f}(n) = G_{d_f,i_f}(n)$. Through using the inequality chain on path from (d_f, i_f) to (d^*, i^*) backwardly, it is obvious that $G_{d^*,i^*}(n) \geq \mu^*$.

Under the condition that event $\mathcal{H}_n$ holds, the *Slack upper bound* value for node (d_n, i_n) could be upper bounded as

$$\begin{aligned} S_{d_n,i_n}(n) &= \bar{p}_{d_n,i_n}(n) + Lm^{d_n} + c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)}{T_{d_n,i_n}(n)}} \\ &\leq \mu\left(b_{d_n,i_n}\right) + c\sqrt{\frac{\ln\frac{\sqrt{4N-2n}}{\varepsilon}}{T_{d_n,i_n}(n)}} + Lm^{d_n} + c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)}{T_{d_n,i_n}(n)}} \\ &\leq \mu\left(b_{d_n,i_n}\right) + Lm^{d_n} + 2c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)}{T_{d_n,i_n}(n)}} \end{aligned} \tag{46}$$

As discussed above, based on the inequality chain, we have that $G_{d,i}(n) \leq G_{d+1,i^*} \leq S_{d+1,i}(n)$. Since $d^* < d_n$, therefore $S_{d_n,i_n}(n) \geq G_{d^*,i^*}(n) \geq \mu^*$, which derives

$$\mu\left(b_{d_n,i_n}\right) + Lm^{d_n} + 2c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)}{T_{d_n,i_n}(n)}} \geq \mu^* \tag{47}$$

Based on above inequality, we could derive the upper bound of Δ_{d_n,i_n} as

$$\begin{aligned} \Delta_{d_n,i_n} &= \mu^* - \mu\left(b_{d_n,i_n}\right) \leq Lm^{d_n} + 2c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)}{T_{d_n,i_n}(n)}} \\ &\leq Lm^{d_n} + 2c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-n}}{\varepsilon}\right)}{T_{d_n,i_n}(n)}} \leq_{(4)} 3c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-n}}{\varepsilon}\right)}{T_{d_n,i_n}(n)}} \end{aligned} \tag{48}$$

The inequality (4) is on basis of the fact that the third term of the *Slack upper bound* equation is no less than the second term in that equation during all operation periods since the node will be partitioned into narrower covering bidding space when $T^*_{d_n} \geq \frac{c^2}{L^2 m^{2d_n}} \ln(\frac{\sqrt{4N-\hat{n}}}{\varepsilon})$. In addition, it is obvious that parent node for (d_n, i_n) is $3Lm^{d_n-1}$-optimal since its selected times exceed $T^*_{d_n-1}$.

Define $n^+_{d,i}$ as the last time or auction round that node (d, i) is selected. Next, the cumulative value of Δ_{d_n,i_n} all over total N periods could be bounded as

$$\begin{aligned} \sum_{n=1}^{N} \Delta_{d_n,i_n} \mathbb{I}\mathcal{H}_n &\leq \sum_{d=0}^{\mathcal{D}(N)} \sum_{i \in \mathcal{A}_d(n)} \sum_{n=1}^{N} \Delta_{d,i} \mathbb{I}\{(d_n, i_n) = (d,i)\} \mathbb{I}\mathcal{H}_n \\ &\leq \sum_{d=0}^{\mathcal{D}(N)} \sum_{i \in \mathcal{A}_d(n)} \sum_{n=1}^{N} 3c\sqrt{\frac{\ln\left(\frac{\sqrt{4N-n}}{\varepsilon}\right)}{T_{d,i}(n)}} \\ &\leq_{(5)} \sum_{d=0}^{\mathcal{D}(N)} \sum_{i \in \mathcal{A}_d(n)} \sum_{s=1}^{T_{d,i}(N)} 3c\sqrt{\frac{2\ln\left(\frac{\sqrt{4N-2n^+_{d,i}}}{\varepsilon}\right)}{s}} \\ &\leq \sum_{d=0}^{\mathcal{D}(N)} \sum_{i \in \mathcal{A}_d(n)} \int_{1}^{T_{d,i}(N)} 3c\sqrt{\frac{2\ln\left(\frac{\sqrt{4N-2n^+_{d,i}}}{\varepsilon}\right)}{s}} ds \\ &\leq \sum_{d=0}^{\mathcal{D}(N)} \sum_{i \in \mathcal{A}_d(n)} 6\sqrt{2}c\sqrt{T_{d,i}(N) \ln\left(\frac{\sqrt{4N-2n^+_{d,i}}}{\varepsilon}\right)} \\ &= 6\sqrt{2}c \sum_{d=0}^{\widetilde{\mathcal{D}}} \sum_{i \in \mathcal{A}_d(n)} \sqrt{T_{d,i}(N) \ln\left(\frac{\sqrt{4N-2n^+_{d,i}}}{\varepsilon}\right)} \\ &\quad + 6\sqrt{2}c \sum_{d=\widetilde{\mathcal{D}}+1}^{\mathcal{D}(N)} \sum_{i \in \mathcal{A}_d(n)} \sqrt{T_{d,i}(N) \ln\left(\frac{\sqrt{4N-2n^+_{d,i}}}{\varepsilon}\right)} \end{aligned} \tag{49}$$

For inequality (5), even for an extreme condition that $n_{d,i}^{+}=N$ while $n\in\{0,1,2,\dots\dots N\}$, we could bound $\ln\left(\sqrt{4N-n}/\varepsilon\right)$ by $2\ln\left(\sqrt{4N-2n_{d_n,i_n}^{+}}/\varepsilon\right)$.

According to the partition principle, the selected times for node (d,i) shall not be larger than $T_d^*(n)$ at round n. Consequently, the first term could be further rewrite as

$$6\sqrt{2}c\sum_{d=0}^{\widetilde{\mathcal{D}}}\sum_{i\in\mathcal{A}_d(n)}\sqrt{T_{d,i}(N)\ln\left(\frac{\sqrt{4N-2n_{d,i}^{+}}}{\varepsilon}\right)}\le$$
$$6\sqrt{2}c\sum_{d=0}^{\widetilde{\mathcal{D}}}\sum_{i\in\mathcal{A}_d(n)}\sqrt{T_{d,i}(N)\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)}\le$$
$$6\sqrt{2}c\sum_{d=0}^{\widetilde{\mathcal{D}}}\mathcal{A}_d(n)\sqrt{T_d^*(N)\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)} \tag{50}$$

Since we adopt a binary tree structure, each node could only generate two children nodes at most. Recall definition defined in Section III-A, the number of $3Lm^{d_n-1}$-optimal nodes is no larger than $C_h(L_1m^{d_n-1})^{-d_c}$. Thus, we have $\mathcal{A}_d(n)\le 2\mathcal{A}_{d-1}^{+}(n)\le 2C_h(L_1m^{d-1})^{-d_c}$. Then

$$6\sqrt{2}c\sum_{d=0}^{\widetilde{\mathcal{D}}}\mathcal{A}_d(n)\sqrt{T_d^*(N)\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)}\le$$
$$6\sqrt{2}c\sum_{d=0}^{\widetilde{\mathcal{D}}}2C_h(L_1m^{d-1})^{-d_c}\sqrt{\frac{c^2}{L^2m^{2d}}\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)}$$
$$\le$$
$$12\sqrt{2}cC_hL_1^{-d_c}\frac{c\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)}{L}m^{d_c}\sum_{d=0}^{\widetilde{\mathcal{D}}}m^{-d(d_c+1)}\le$$
$$12\sqrt{2}cC_hL_1^{-d_c}\frac{c\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)}{L}m^{d_c}\frac{m^{-\widetilde{\mathcal{D}}(d_c+1)}}{1-m} \tag{51}$$

After acquiring the upper bound of the first term, the second term is further bounded as

$$6\sqrt{2}c\sum_{d=\widetilde{\mathcal{D}}+1}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d(n)}\sqrt{T_{d,i}(N)\ln\left(\frac{\sqrt{4N-2n_{d,i}^{+}}}{\varepsilon}\right)}\le_{(6)}$$
$$6\sqrt{2}c\sqrt{\sum_{d=\widetilde{\mathcal{D}}+1}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d(n)}\ln\left(\frac{\sqrt{4N-2n_{d,i}^{+}}}{\varepsilon}\right)}\sqrt{\sum_{d=\widetilde{\mathcal{D}}+1}^{\mathcal{D}(n)}\sum_{i\in\mathcal{A}_d(n)}T_{d,i}(N)}$$
$$\le_{(7)}6\sqrt{2}c\sqrt{\sum_{d=\widetilde{\mathcal{D}}+1}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d(n)}\ln\left(\frac{\sqrt{4N-2n_{d,i}^{+}}}{\varepsilon}\right)}\sqrt{N} \tag{52}$$

where inequality (6) is derived from the Cauchy-Schwarz inequality and inequality (7) is based on the fact that $N=\sum_{d=0}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d(n)}T_{d,i}(n)$. We further derive a sequence of inequalities

$$N=\sum_{d=0}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d(n)}T_{d,i}(n)\ge\sum_{d=\widetilde{\mathcal{D}}+1}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d^{+}(n)}T_d^*(n)$$
$$\ge\sum_{d=\widetilde{\mathcal{D}}}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d^{+}(n)}\frac{c^2}{L^2m^{2d}}\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)$$
$$\ge\frac{c^2m^{-2\widetilde{\mathcal{D}}}}{L^2}\sum_{d=\widetilde{\mathcal{D}}}^{\mathcal{D}(N)}m^{2(\widetilde{\mathcal{D}}-d)}\sum_{i\in\mathcal{A}_d^{+}(n)}\ln\left(\frac{\sqrt{4N-\hat{n}}}{\varepsilon}\right)$$
$$\ge\sum_{d=\widetilde{\mathcal{D}}}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d^{+}(n)}\frac{c^2}{L^2m^{2\widetilde{\mathcal{D}}}}\ln\left(\frac{\sqrt{4N-2n_{h,i}^{+}}}{\varepsilon}\right) \tag{53}$$

Inverting above inequality, we could obtain

$$\frac{NL^2m^{2\widetilde{\mathcal{D}}}}{c^2}\ge\sum_{d=\widetilde{\mathcal{D}}}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d^{+}(n)}\ln\left(\frac{\sqrt{4N-2n_{h,i}^{+}}}{\varepsilon}\right) \tag{54}$$

Then we could further derive that

$$6\sqrt{2}c\sum_{d=\widetilde{\mathcal{D}}}^{\mathcal{D}(N)}\sum_{i\in\mathcal{A}_d(n)}\sqrt{T_{d,i}(N)\ln\left(\frac{\sqrt{4N-2n_{h,i}^{+}}}{\varepsilon}\right)}$$
$$\le 6\sqrt{2}c\sqrt{\frac{NL^2m^{2\widetilde{\mathcal{D}}}}{c^2}}\sqrt{N}=6\sqrt{2}NLm^{\widetilde{\mathcal{D}}} \tag{55}$$

Based on these inequalities, the upper bound of $\sum_{n=1}^{N}\Delta_{d_n,i_n}\mathbb{I}\mathcal{H}_n$ could be further formulated as

$$\sum_{n=1}^{N}\Delta_{d_n,i_n}\mathbb{I}\mathcal{H}_n\le$$
$$12\sqrt{2}cC_hL_1^{-d_c}\frac{c\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)}{L}m^{d_c}\frac{m^{-\widetilde{\mathcal{D}}(d_c+1)}}{1-m}+6\sqrt{2}NLm^{\widetilde{\mathcal{D}}} \tag{56}$$

Obviously, through delicately selecting the value of $\widetilde{\mathcal{D}}$, we could tightly bound the upper bound of $\sum_{n=1}^{N}\Delta_{d_n,i_n}\mathbb{I}\mathcal{H}_n$. Therefore, using the basic inequality to make two terms in above inequality equivalent enables to acquire optimal value of $\widetilde{\mathcal{D}}$.

$$12\sqrt{2}cC_hL_1^{-d_c}\frac{c\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)}{L}m^{d_c}\frac{m^{-\widetilde{\mathcal{D}}(d_c+1)}}{1-m}=6\sqrt{2}NLm^{\widetilde{\mathcal{D}}}$$
$$\Rightarrow m^{\widetilde{\mathcal{D}}}$$
$$=\left(\frac{2c^2C_hL_1^{-d_c}\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)m^{d_c}}{NL^2(1-m)}\right)^{\frac{1}{d_c+2}} \tag{57}$$

Based on the $\widetilde{\mathcal{D}}$ value, we have that

$$\sum_{n=1}^{N}\Delta_{d_n,i_n}\mathbb{I}\mathcal{H}_n\le$$
$$2\left(12\sqrt{2}cC_hL_1^{-d_c}\frac{c\ln\left(\frac{2\sqrt{N}}{\varepsilon}\right)}{L}m^{d_c}\frac{m^{-\widetilde{\mathcal{D}}(d_c+1)}}{1-m}*6\sqrt{2}NLm^{\widetilde{\mathcal{D}}}\right)^{1/2}$$
$$=24\sqrt{2}\left(\frac{L^{d_c}c^2C_hL_1^{-d_c}m^{d_c}}{1-m}\right)^{\frac{1}{d_c+2}}\log\left(\frac{2\sqrt{N}}{\varepsilon}\right)^{\frac{1}{d_c+2}}N^{\frac{d_c+1}{d_c+2}} \tag{58}$$

Note that, the near-optimality dimension d_c can be much smaller than the actual dimension of arm space (bidding space $\mathcal{B}$ here) in continuous case and Bubeck et al [23] proved that the near-optimality dimension could equal to 0 under some mild smoothness assumption. Therefore, the result shows that our algorithm guarantees sub-linear regret in continuous arm space.

APPENDIX F

The computational complexity of the proposed TBS algorithm is $\mathcal{O}(n \log n)$ due to the growth rate of tree's depth and the procedure's structure. By *Remark 1*, we know that the maximum depth $\mathcal{D}(n)$ of the tree $\mathcal{T}_n$ is $\mathcal{O}(\log n)$. Therefore, at round n, the computational cost of traversing the optimal path P_n to find the optimal b_{d_n,i_n} is at most $\mathcal{O}(\log n)$, which also includes cost of calculating $S_{d,i}(n)$ and $G_{d,i}(n)$ for nodes in P_n. Such linearithmic time complexity makes TBS algorithm an effective method for *big-data* applications.