Generalized Online Auctions with Time Varying Values

Abstract

Online auctions mechanism design for re-usable goods has wide application fields, such as network storage distribution, computing resources allocation. Agent′s types, especially values, may vary while they are waiting for being allocated. But the variation of agent′s values are rarely discussed. We consider the mechanism design for allocation of online resources between agents whose values vary with time. After extending the classic pricing strategy, we propose a generalized mechanism which resists bid-cheating and derive the upper bound of competitive ratio within constant factors with the offline optimal solution. We also prove that this mechanism remains effective, despite of the misreport of the arrival and departure time, in the scenario of decreasing values.

1 Introduction

The online mechanism is aimed for settings where multiple agents arrive and depart overtime, the allocation decisions and payment determinations have to be made on the fly without any knowledge of future arrivals.

Solutions to many interesting decision problems in dynamic multi-agent environments have referred to the online mechanism design, e.g., selling seats on an airplane to buyers arriving over time and allocating computational resources (storage space, CPU, etc.). [Gerding *et al.*, 2011] designed a novel online auction protocol for electric vehicle charging. [Shi *et al.*, 2014] discussed an online combinatorial auction designed in the cloud computing paradigm.

In most designs of the online mechanism, authors treat the values of agents as constants over time, or for simplicity, they only consider unit job length and unit requirement, [Wu *et al.*, 2014]. However, in practice, these strict restrictions are rarely satisfied. For instance, in online cloud computing resources allocation, agents require multi-unit of storage resources to finish the job, and the requirement continues multiple time slots. Time variations also exist. The degree of emergency usually decreases with time passing, and causes the agent's value to drop. Therefore, establishing a generalized online auction model taking into account of the above factors is highly necessary.

In this paper, we propose a generalized model for online auctions, taking into account of agents' varying values over time (special but important scenarios, such as decreasing values and constant values, are included.) In the model, each agent reports to the auctioneer its type and and initial value which may vary over time. The varying factor function of each agent is assumed to be public knowledge and not included in the reported type. And for simplicity of analysis, we do not consider the possibility of misreport on number of instances and job length. We leave them for the future work.

Our objective is to design a mechanism for the model which ensures strategy-proofness not only for values, but also for the arrival and departure time. [Lavi and Nisan, 2000] proved that without any restriction on the types of possible misreports, it is impossible to achieve a bounded competitive ratio. Thus, in our model, we naturally eliminate the possibility of reports of early arrivals and late departures.

We present a new online mechanism for the proposed generalized model. For the allocation stage, we adopt dynamic programming which is computationally efficient. For the pricing stage, instead of taking the minimum of the critical prices over one period, we naturally extend Myerson's Lemma and use the dominant strategy constraint to narrow down the payment to a single candidate.

The paper is organized as follows. Section 2 introduces the model and the basic background knowledge. Section 4 focuses on the detailed design of our mechanism. In the first part of Section 4, we explain the process of virtual bid generation. Then, we introduce the dynamic programming allocation method designed for multiple goods allocation and then give an analysis for the upper bound of competitive ratio. In the third part of Section 4, we introduce our extension on the classic lemma which helps to derive the unique price, which ensures truthfulness on bids report. In Section 5, we give the simulation results on social welfare. In Section 6, we review the related works. Finally, we conclude with some future directions in Section 7.

2 Preliminaries

In this section, we introduce the model of online auction with time varying values, and briefly review the related solution concepts used in this paper from game theory.

2.1 Auction Model

We consider the problem of online auction mechanism design for trading M re-usable homogeneous items over a finite time interval. We further divide the time interval into T equal length slots: $\mathbb{T} = \{1, 2, \cdots, T\}$. There are n agents $\mathbb{N} = \{1, 2, \cdots, N\}$, and they arrive and leave the auction in a random order. Agent $i \in \mathbb{N}$ arrives at the auction at a_i, and demands for w_i items for l_i continuous time slots to execute her job before the deadline (or departure time) d_i. The valuation function of the agent i is $v_i(t)$, meaning that the agent has valuation $v_i(t)$ on the job, once she obtains w_i allocated items at time t, and last for the coming consecutive l_i slots. We also denote the intrinsic valuation by v_i, *i.e.*, $v_i = v_i(a_i)$. The job request of the agent i can be expressed as a vector of five elements: $\theta_i = (a_i, d_i, m_i, l_i, v_i)$, which is also called as type in mechanism design. Let vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_N)$ denote the types of all the agents. In the time varying value scenario, the agent i's valuation function can be expressed as:

$$v_i(t) = \begin{cases} \max(v_i \times f_i(t) + h_i(t), 0), & t \in [a_i, d_i - l_i], \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where $f_i(t)$ and $h_i(t)$ are multiplication and addition varying functions, respectively. Different from the valuation model in [Wu *et al.*, 2014] , we do not restrict $f_i(t)$ and $h_i(t)$ to any specific formats, but only require that $f_i(t)$ should be monotone and non-decreasing, which is the necessary condition for the existence of the strategy-proof online auction mechanism in the time varying value setting. We illustrate some representative valuation functions as follows:

- Exponential Varying: $f_i(t) = \eta^{(t-a_i)}$, where $\eta \in (0, 1)$, and $h_i(t) = \beta$.
- Quadratic Varying: $f_i(t) = \beta, h_i(t) = \delta(t - \frac{a_i+d_i}{2})^2$.
- Joint Varying: $f_i(t) = \eta^{(t-a_i)}$, $h_i(t) = \delta(t - \frac{a_i+d_i}{2})^2$, where $\eta \in (0, 1)$.
- No Varying: $f_i(t) = 1, h_i(t) = 0$.

In this paper, we assume that the varying parameters η, β and δ are public knowledge, and the agents share common varying parameters, and then the valuation function, which is similar to the assumption made in paper []. We further assume that the valuation gap is bound by κ, *i.e.*, $\max_{a_i \le t \le d_i - l_i} v_i(t) / \min_{a_i \le t \le d_i - l_i} v_i(t) \le \kappa$.

Once an agent $i \in \mathbb{N}$ enters into the online auction, she submits a declared type $\hat{\theta}_i = (\hat{a}_i, \hat{d}_i, m_i, l_i, \hat{v}_i)$, which may not be necessarily equal to her type θ_i, to a trusted auctioneer. In this paper, we do not design truthful online auction to resist all possible misreports, because it has been proven that to resist any type of misreport, an auction may suffer serve degradation in revenue and efficiency [Hajiaghayi, 2005]. we focus on the scenario, in which the selfish and rational agent can cheat the arrival time a_i, departure time d_i, as well as the intrinsic valuation v_i during the bidding process. In contrast to these misreporting patterns, we argue that the agent has less incentive to misreport the number of demanded items m_i and the job length l_i. By requesting for more items or longer job length, the agent has to pay more than necessary to satisfy her job request. When asking for shorter job length or less items, the agent cannot finish her job. In addition, we restrict that the agents can only report late arrival times and early departure times, *e.g.*, $\hat{a}_i \ge a_i$ and $\hat{d}_i \le d_i$, because we can prevent the agents from reporting early arrival and late departure by adopting the heart-beat scheme in []. We use vector $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_N)$ to denote the declared types of all agents.

At each time slot $t \in \mathbb{T}$, the auctioneer first calculates the bid $b_i(t)$ for each active agent $i \in \mathbb{N}_a$, whose departure time is larger than the current time t, *i.e.*, $\hat{a}_i < t < \hat{d}_i$, by using her declared information $\hat{\theta}_i$ and the equation (Please refer to Section 4 for the detailed discussion on the bid calculation.). Given the bidding profile of the active agents $\mathbb{N}_a$ at time t: $\mathbf{b}(t) = (b_1(t), b_2(t), \cdots, b_{|\mathbb{N}_a|}(t))$, the auctioneer then allocates M items, including idle items and those in used by working jobs, to the active agents. We note that the new arriving agents with high bids may interrupt some working jobs. We call the agent $i \in \mathbb{N}$ as a winner when her job runs for l_i continuous time slots before the deadline $\hat{d}_i$ without any interruption. We use $x_i(\hat{\boldsymbol{\theta}}) = 1$ to denote that the agent i is a winner when the declare information is $\hat{\boldsymbol{\theta}}$; otherwise $x_i(\hat{\boldsymbol{\theta}}) = 0$. Finally, according to the declared information $\hat{\boldsymbol{\theta}}$ of agents, the auctioneer determines the payment $p_i(\hat{\boldsymbol{\theta}})$ for each agent i at her declared departure time $\hat{d}_i$. The payments to the losing agents are set to zeros. We use vector $\mathbf{x}(\hat{\boldsymbol{\theta}}) = (x_1(\hat{\boldsymbol{\theta}}), x_2(\hat{\boldsymbol{\theta}}), \cdots, x_N(\hat{\boldsymbol{\theta}}))$ and $\mathbf{p}(\hat{\boldsymbol{\theta}}) = (p_1(\hat{\boldsymbol{\theta}}), p_2(\hat{\boldsymbol{\theta}}), \cdots, p_N(\hat{\boldsymbol{\theta}}))$ to represent the allocation rule and payment rule in an online auction, respectively.

The utility u_i of each agent $i \in \mathbb{N}$ is defined as the difference between her valuation on the allocated items and the payment:

$$u_i(\hat{\boldsymbol{\theta}}) = \begin{cases} v_i(t_i(\hat{\boldsymbol{\theta}})) \times x_i(\hat{\boldsymbol{\theta}}) - p_i(\hat{\boldsymbol{\theta}}), & i \in \mathbb{W}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $\mathbb{W}$ is the set of winning agents, and $t_i(\hat{\boldsymbol{\theta}})$ is the allocation time of the winner $i \in \mathbb{W}$ when the declared information profile is $\hat{\boldsymbol{\theta}}$.

In this paper, the agents are rational and selfish, and always want maximize their utilities by strategically reporting their private information. In contrast to this selfish goals, the objective of the auctioneer is to maximize *social welfare*, which is defined as follows.

Definition 1 (Social Welfare)**.** *The social welfare in an online auction with time varying value is the sum of winners' valuation over the allocated items at their corresponding winning time slots,* i.e.,

$$SW = \sum_{i \in \mathbb{W}} v_i(t_i(\hat{\boldsymbol{\theta}})). \tag{3}$$

2.2 Solution Concepts

A strong solution concept from mechanism design is *dominant strategy*.

Definition 2 (Dominant Strategy)**.** *Strategy θ_i is agent i's dominant strategy, if for any strategy $\hat{\theta}_i \neq \theta_i$ and any other player's strategy profile $\boldsymbol{\theta}_{-i}$, we have*

$$u_i(\theta_i, \boldsymbol{\theta}_{-i}) \geq u_i(\hat{\theta}_i, \boldsymbol{\theta}_{-i}). \tag{4}$$

Intuitively, a dominant strategy of a player is a strategy that maximizes her utility, regardless of what strategy profile the other players choose.

The solution to the afore mentioned online auction is a kind of *direct revelation mechanism*, in which the strategies of the agents are to directly propose bids based on their types. The concept of dominant strategy is the basis of *incentive-compatible* direct revelation mechanism, which means that revealing truthful information is a dominant strategy for every player. An accompanying concept is *individual-rationality*, which means that every player participating in the game expects to gain no less utility than staying outside. We now can introduce the definition of *strategy-proof direct revelation mechanism*.

Definition 3 (Strategy-Proof Direct Revelation Mechanism)**.** *A direct revelation mechanism is strategy-proof, when it satisfies both incentive-compatibility and individual-rationality.*

The objective of this work is to design strategy-proof online auction mechanisms in time varying values setting.

3 Characterizing Strategy-Proofness

In this section, we present a characterization theorem for strategy-proof online mechanisms with time varying values. This can be considered as a generalization of the well-known Myerson theorem for the truthful mechanisms with one-parameter agents. Specifically, we claim that the necessary and sufficient condition for a payment rule that truthfully implement an allocation rule in time varying values setting is that the function $F(\boldsymbol{\theta}) = f(t(\boldsymbol{\theta})) \times x(\boldsymbol{\theta})$ must satisfy the monotonicity criterion. We first give the definition of this monotone criterion.

Definition 4 (Monotonicity)**.** *In an online auction, we say a type $\theta_i = (a_i, d_i, v_i)$ dominates another type $\hat{\theta}_i = (\hat{a}_i, \hat{d}_i, \hat{v}_i)$, denoted by $\theta_i \succ \hat{\theta}_i$, if $a_i \leq \hat{a}_i$, $d_i \geq \hat{d}_i$ and $v_i > \hat{v}_i$. The function $F_i(\boldsymbol{\theta}) = f_i(t_i(\boldsymbol{\theta})) \times x_i(\boldsymbol{\theta})$ is monotone, if for every agent i, $\theta_i \succ \hat{\theta}_i$ and the reported types of the other agent $\boldsymbol{\theta}_{-i}$, we have $F_i(\theta_i, \boldsymbol{\theta}_{-i}) \geq F_i(\hat{\theta}_i, \boldsymbol{\theta}_{-i})$.*

We now present our major result: the necessary and sufficient condition for the existence of strategy-proof online auctions with time varying values.

Theorem 1. *There exist a payment rule $\mathbf{p}(\boldsymbol{\theta})$ such that the online auction $(\mathbf{x}(\boldsymbol{\theta}), \mathbf{p}(\boldsymbol{\theta}))$ in time varying values setting is strategy-proof if and only if the function $F_i(\boldsymbol{\theta}) = f_i(t_i(\boldsymbol{\theta})) \times x_i(\boldsymbol{\theta})$ is monotone for each agent $i \in \mathbb{N}$.*

Proof. To simplify the notations, we introduce $H_i(\boldsymbol{\theta}) = h_i(t_i(\boldsymbol{\theta})) \times x_i(\boldsymbol{\theta})$ for each agent $i \in \mathbb{N}$. We first prove the "if" part. Let $F_i(\boldsymbol{\theta})$ be the monotone function for each agent $i \in \mathbb{N}$ and a type vector $\boldsymbol{\theta} = (\theta_i, \boldsymbol{\theta}_{-i})$, where $\theta_i = (a_i, d_i, v_i)$. We set the payment rule as

$$p_i(\boldsymbol{\theta}) = \sum_{k=1}^{K} \left[v_i^k \Delta_i^F(a_i, d_i, v_i^k) + \Delta_i^H(a_i, d_i, v_i^k) \right], \tag{5}$$

The sequence $v_i^0, v_i^1, v_i^2, \cdots, v_i^K$ is a list of K valuations, which are the break points of function $F_i(\boldsymbol{\theta})$ or $H_i(\boldsymbol{\theta})$. In general, we assume $v_i^{k_1} \leq v_i^{k_2}$ for $k_1 \leq k_2$, $v_i^0 = 0$ and $v_i^K \leq v_i$. The function $\Delta_i^F(a_i, d_i, v_i^k)$ (or $\Delta_i^H(a_i, d_i, v_i^k)$) represents the jump of $F_i(\boldsymbol{\theta})$ (or $H_i(\boldsymbol{\theta})$) at $(\theta_i^k, \boldsymbol{\theta}_{-i})$, where $\theta_i^k = (a_i, d_i, v_i^k)$, *i.e.*,

$$\Delta_i^F(a_i, d_i, v_i^k) = F_i(\theta_i^k, \boldsymbol{\theta}_{-i}) - F_i(\theta_i^{k-1}, \boldsymbol{\theta}_{-i}), \forall 1 \leq k \leq K.$$
$$\Delta_i^H(a_i, d_i, v_i^k) = H_i(\theta_i^k, \boldsymbol{\theta}_{-i}) - H_i(\theta_i^{k-1}, \boldsymbol{\theta}_{-i}), \forall 1 \leq k \leq K.$$

Combining the definition of utility function with the payment rule, we can express the utility $u_i(\boldsymbol{\theta})$ of aggent $i \in \mathbb{N}$ as:

$$\begin{aligned}
u_i(\boldsymbol{\theta}) &= [v_i \times f_i(t(\boldsymbol{\theta})) + h_i(t(\boldsymbol{\theta}))] \times x(\boldsymbol{\theta}) \\
&\quad - \sum_{k=1}^{K} \left[v_i^k \Delta_i^F(a_i, d_i, v_i^k) + \Delta_i^H(a_i, d_i, v_i^k) \right] \\
&= v_i \times f_i(t(\boldsymbol{\theta})) \times x(\boldsymbol{\theta}) - \sum_{k=1}^{K} v_i^k \Delta_i^F(a_i, d_i, v_i^k) \\
&= v_i \times F_i(\boldsymbol{\theta}) - \sum_{k=1}^{K} v_i^k \Delta_i^F(a_i, d_i, v_i^k) \\
&= \left(v_i^K + v_i - v_i^K \right) \times F_i\left(\theta_i^K, \theta_{-i}\right) \\
&\quad - \sum_{k=1}^{K} v^k \times \left[F_i\left(\theta_i^k, \theta_{-i}\right) - F_i(\theta_i^{k-1}, \theta_{-i}) \right] \\
&= \left(v_i - v_i^K \right) \times F_i\left(\theta_i^K, \theta_{-i}\right) \\
&\quad + \sum_{k=1}^{K} \left(v_i^k - v_i^{k-1} \right) \times F_i\left(\theta_i^{k-1}, \theta_{-i}\right)
\end{aligned} \tag{6}$$

According to the definition of the valuation sequence, we have $v_i^K \leq v_i$ and $v_i^{k-1} \leq v_i^k$ for all $1 \leq k \leq K$. Therefore, the utility $u_i(\boldsymbol{\theta})$ of aggent i can not be negative, which satisfy the property of *Individual Rationale*.

We now show that the monotone function $F_i(\boldsymbol{\theta})$ in combination with the payment rule $p_i(\boldsymbol{\theta})$ guarantee the property of *Incentive Compatibility*. We prove this by contradiction. If the mechanism is not incentive compatible, there is aggent i, a true type $\theta_i = (a_i, d_i, v_i)$, and a non-truthful reported type $\hat{\theta}_i = \left(\hat{a}_i, \hat{d}_i, \hat{v}_i\right)$ with $\hat{a}_i \geq a_i$, $\hat{d}_i \leq d_i$ and $\hat{v}_i \neq v_i$, such that the utility $\hat{u}_i\left(\hat{\theta}_i, \boldsymbol{\theta}_{-i}\right)$ of aggent i if she reports $\hat{\theta}_i$ is strictly greater than the utility $u_i(\theta_i, \boldsymbol{\theta}_{-i})$ that she can achieve from being truthful, *i.e.*, $\hat{u}_i\left(\hat{\theta}_i, \boldsymbol{\theta}_{-i}\right) > u_i(\theta_i, \boldsymbol{\theta}_{-i})$. By Equation (6), we have

$$\left(v_i - v^{\widehat{K}_i}\right) F_i\left(\hat{\theta}_i^{\widehat{K}}, \boldsymbol{\theta}_{-i}\right) + \sum_{k=1}^{\widehat{K}} \left(v_i^k - v_i^{k-1}\right) F_i\left(\hat{\theta}_i^{k-1}, \boldsymbol{\theta}_{-i}\right)$$
$$> \left(v_i - v_i^K\right) F_i\left(\theta_i^K, \boldsymbol{\theta}_{-i}\right) + \sum_{k=1}^{K} \left(v_i^k - v_i^{k-1}\right) F_i\left(\theta_i^{k-1}, \boldsymbol{\theta}_{-i}\right) \quad (7)$$

According to the monotonicity of the function $F_i(\boldsymbol{\theta})$ and the inequalities $\hat{a}_i \geq a_i$ and $\hat{d}_i \leq d_i$, we have the following relation:

$$\begin{aligned}
&\textbf{RHS of } (7) = \left(v_i - v_i^{\widehat{K}}\right) \times F_i\left(\left(\hat{a}_i, \hat{d}_i, v_i^{\widehat{K}}\right), \boldsymbol{\theta_{-i}}\right) \\
&\quad + \sum_{k=1}^{\widehat{K}} (v_i^k - v_i^{k-1}) \times F_i\left(\left(\hat{a}_i, \hat{d}_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \\
\leq \quad & \left(v_i - v_i^{\widehat{K}}\right) \times F_i\left(\left(a_i, d_i, v_i^{\widehat{K}}\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \sum_{k=1}^{\widehat{K}} (v_i^k - v_i^{k-1}) \times F_i\left(\left(a_i, d_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \qquad (8)
\end{aligned}$$

Equations (7) and (8) imply that **RHS of** (8) is greater than **LHS of** (7), *i.e.*,

$$\begin{aligned}
&\left(v_i - v_i^{\widehat{K}}\right) \times F_i\left(\left(a_i, d_i, v_i^{\widehat{K}}\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \sum_{k=1}^{\widehat{K}} (v_i^k - v_i^{k-1}) \times F_i\left(\left(a_i, d_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \\
> \quad & \left(v_i - v_i^K\right) \times F_i\left(\left(a_i, d_i, v_i^K\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \sum_{k=1}^{K} \left(v_i^k - v_i^{k-1}\right) F_i\left(\left(a_i, d_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \qquad (9)
\end{aligned}$$

We complete the analysis by distinguishing two cases:

• If $\hat{v}_i < v_i$, we then have $\widehat{K} \leq K$, and thus $v_i^{\widehat{K}} \leq v_i^K$. Since the function $F_i(\boldsymbol{\theta})$ is monotone with respective to v_i^k, we can get:

$$\begin{aligned}
&\textbf{RHS of } (9) \geq \left(v_i - v_i^K\right) \times F_i\left(\left(a_i, d_i, v_i^{\widehat{K}}\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \sum_{k=\widehat{K}+1}^{K} \left(v_i^k - v_i^{k-1}\right) \times F_i\left(\left(a_i, d_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \sum_{k=1}^{\widehat{K}} (v_i^k - v_i^{k-1}) \times F_i\left(\left(a_i, d_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \\
\geq \quad & (v_i - v_i^{\widehat{K}}) \times F_i\left(\left(a_i, d_i, v_i^{\widehat{K}}\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \sum_{k=1}^{\widehat{K}} (v_i^k - v_i^{k-1}) \times F_i\left(\left(a_i, d_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \\
= \quad & \textbf{LHS of } (9)
\end{aligned}$$

Therefore, we get a contradiction in this case.

• If $\hat{v}_i > v_i$, we then have $\widehat{K} \geq K$, and thus $v_i^{\widehat{K}} \geq v_i^K$. By the monotonicity of the function $F_i(\boldsymbol{\theta})$, we have

$$\begin{aligned}
&\textbf{LHS of } (9) \leq \left(v_i - v_i^{K+1} + \sum_{k=K+2}^{\widehat{K}} (v_i^{k-1} - v_i^k)\right) \\
&\quad \times F_i\left(\left(a_i, d_i, v_i^{\widehat{K}}\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \sum_{k=1}^{K} \left(v_i^k - v_i^{k-1}\right) F_i\left(\left(a_i, d_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \left(v_i - v_i^K + v_i^{K+1} - v_i\right) \times F_i\left(\left(a_i, d_i, v_i^K\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \sum_{k=K+2}^{\widehat{K}} \left(v_i^k - v_i^{k-1}\right) F_i\left(\left(a_i, d_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \\
\leq \quad & \left(v_i - v_i^K\right) \times F_i\left(\left(a_i, d_i, v_i^K\right), \boldsymbol{\theta}_{-i}\right) \\
&\quad + \sum_{k=1}^{K} \left(v_i^k - v_i^{k-1}\right) F_i\left(\left(a_i, d_i, v_i^{k-1}\right), \boldsymbol{\theta}_{-i}\right) \\
= \quad & \textbf{RHS of } (9)
\end{aligned}$$

Thus, we also get a contradiction in this cases. We completed the proof of the "if" part.

Conversely, we now consider the "only if" part, and assume that $F_i(\boldsymbol{\theta})$ is the function, for which there is a payment rule $\boldsymbol{p}(\boldsymbol{\theta})$ such that $(\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{p}(\boldsymbol{\theta}))$ is strategy-proof. Consider a aggent $i \in \mathbb{N}$ and two types $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}$ with $\boldsymbol{\theta}_{-i} = \hat{\boldsymbol{\theta}}_{-i}$ and $\theta_i \succ \hat{\theta}_i$. We first consider a scenario where the true types of the agents is $\boldsymbol{\theta}$. The strategy-proof mechanism ensures that the utility of the aggent i when bidding truthfully is not less than that when she misreports her type, *i.e.*,

$$\begin{aligned}
&[f_i(t_i(\boldsymbol{\theta})) \times v_i + h_i(t_i(\boldsymbol{\theta}))] \times x_i(\boldsymbol{\theta}) - p_i(\boldsymbol{\theta}) \\
\geq \quad & \left[f_i(t_i(\hat{\boldsymbol{\theta}})) \times v_i + h_i(t_i(\hat{\boldsymbol{\theta}}))\right] \times x_i(\hat{\boldsymbol{\theta}}) - p_i(\hat{\boldsymbol{\theta}}). \quad (10)
\end{aligned}$$

We then consider another scenario where the true type of the aggent i is $\hat{\theta}_i$ and she may cheat by misreporting θ_i. Similarly, we have

$$\begin{aligned}
&\left[f_i(t_i(\hat{\boldsymbol{\theta}})) \times \hat{v}_i + h_i(t_i(\hat{\boldsymbol{\theta}}))\right] \times x_i(\hat{\boldsymbol{\theta}}) - p_i(\hat{\boldsymbol{\theta}}) \\
\geq \quad & [f_i(t_i(\boldsymbol{\theta})) \times \hat{v} + h_i(t_i(\boldsymbol{\theta}))] \times x_i(\boldsymbol{\theta}) - p_i(\boldsymbol{\theta}) \qquad (11)
\end{aligned}$$

Combining Equations (10) and (11), we can get

$$\begin{aligned}
&[f_i(t_i(\boldsymbol{\theta})) \times v_i + h_i(t_i(\boldsymbol{\theta}))] \times x_i(\boldsymbol{\theta}) \\
&- \left[f_i(t_i(\hat{\boldsymbol{\theta}})) \times v_i + h_i(t_i(\hat{\boldsymbol{\theta}}))\right] \times x_i(\hat{\boldsymbol{\theta}}) \\
\geq \quad & p_i(\boldsymbol{\theta}) - p_i(\boldsymbol{\theta}') \\
\geq \quad & [f_i(t_i(\boldsymbol{\theta})) \times \hat{v}_i + h_i(t_i(\boldsymbol{\theta}))] \times x_i(\boldsymbol{\theta}) \\
&- \left[f_i(t_i(\hat{\boldsymbol{\theta}})) \times \hat{v}_i + h_i(t_i(\hat{\boldsymbol{\theta}}))\right] \times x_i(\hat{\boldsymbol{\theta}}) \\
\Rightarrow \quad & f_i(t_i(\boldsymbol{\theta}))x_i(\boldsymbol{\theta}) \times (v_i - \hat{v}_i) \geq f_i(t_i(\hat{\boldsymbol{\theta}}))x(\hat{\boldsymbol{\theta}}) \times (v_i - \hat{v}_i) \\
\Rightarrow \quad & F_i(\boldsymbol{\theta}) \times (v_i - \hat{v}_i) \geq F_i(\hat{\boldsymbol{\theta}}) \times (v_i - \hat{v}_i)
\end{aligned}$$

Since $\theta_i \succ \hat{\theta}_i$, we have $v_i > \hat{v}_i$, and thus $F_i(\boldsymbol{\theta}) \geq F_i(\hat{\boldsymbol{\theta}})$. Therefore, we can conclude that $F_i(\boldsymbol{\theta})$ is monotone.

□

4 Auction Design

In this section, we present the detailed design for the online auction with time varying values, and analyze its economic property, computational complexity and competitive ratio. Our designed mechanism consists of three major components: virtual bid generation, item allocation, and payment calculation.

4.1 Virtual Bid Generation

When a newly arrived agent proposes a bid higher than that of the current working agents, the auctioneer can choose to preempt the working jobs to make up the price difference or reject the request of the new agents to guarantee the fairness of allocation. It has been shown that the preemption may degrade the resource utilization, damage auction credibility, and can potentially discourage agents from participating in future auction [Deek *et al.*, 2011]. Therefore, the auctioneer should preempt a working agents only if the newly arrived agents offer a significantly high bid.

The auctioneer artificially raises the bids of ongoing agents, denoted by $\mathbb{N}_o$, to give them priority during item allocation. Specifically, for an ongoing agent $i \in \mathbb{N}_o$, who requests m_i items for l_i continuous time slots and has been allocated the items at time slot $t_i(\hat{\boldsymbol{\theta}})$, the auctioneer adjusts her bid as $b_i(t)$ at time slot t:

$$b_i(t) = \hat{v}_i(t_i(\hat{\boldsymbol{\theta}})) \times \alpha^{\varphi_i}, \tag{12}$$

where $\hat{v}_i(t_i(\hat{\boldsymbol{\theta}}))$ is similar to the valuation function (Equation (**??**)) by replacing v_i with $\hat{v}_i$. $\varphi_i = (t - t_i(\hat{\boldsymbol{\theta}}))/l_i$ represents i's job completeness at time t, and $\alpha \geq 1$ is the threshold that the auctioneer can adjust to control the preemption frequency. $\alpha = 1$ maps to the classical preemption model. The auctioneer can give more protection to the ongoing agents by increasing α. When $\alpha \to \infty$, the auctioneer does not allow preemption, and the agents can work for requested continuous time slots once they are allocated items.

For the active agents that have not been allocated items, *i.e.*, agents in $\mathbb{N}_a \backslash \mathbb{N}_o$, the auctioneer updates their bids according to the valuation function $\hat{v}_i(t)$ and their reported types $\hat{\boldsymbol{\theta}}$. At time slot $t \in \mathbb{T}$, the virtual bid for the active agent $i \in \mathbb{N}_a \backslash \mathbb{N}_o$ is $b_i(t) = \hat{v}_i(t)$.

The auctioneer can generate the virtual bid $b_i(t)$ of the agent $i \in \mathbb{N}$ at time slot $t \in \mathbb{T}$ by distinguishing the three cases.

$$b_i(t) = \begin{cases} \hat{v}_i(t_i(\hat{\boldsymbol{\theta}})) \times \alpha^{\varphi_i}, & i \in \mathbb{N}_o, \\ \hat{v}_i(t), & i \in \mathbb{N}_a \backslash \mathbb{N}_o, \\ 0, & \text{otherwise}, \end{cases}$$

4.2 Item Allocation

Theorem 2. *The competitive ratio of the item allocation algorithm is*

Proof. We prove this theorem by a charging argument. We charge the value of any winning agent in an optimal solution OPT to a winning agent in our algorithm. For the winning agent i who is allocated items at time t_i in our algorithm, we calculate the maximum charge to her. We further distinguish the following cases.

▶ If agent i is also a winning agent in OPT, then there is a charge $v_i(t_i^*)$ to herself. Here t_i^* is the time slot that the agent i obtains the items in OPT solution. We can bound this charge by $v_i(t_i^*) \leq v_i(t_i) \times \kappa$.

▶ If agent i does not pick by the OPT, we then consider the agents in OPT who is directly/indirectly preempted by the agent i, and denote this set of agents by $\mathbb{N}_i$. We consider that the agent $j \in \mathbb{N}_i$ is preempted by agent j_1, who may be further preempted by another agent. We continue this chain until we reach the agent i who is not preempted.

In the analysis of competitive ratio, we use v_i^t instead of b_i^t(agent i's value at time t). The key idea is to charge the values of winning agents in the optimal solution OPT to winning agents in our allocation algorithm.

For agent j who wins at time t_j in our allocation algorithm, we find the maximum value charged to her.

- Firstly, we consider the case that j also wins in the OPT, then there's a charge $v_j^{t_{opt}}$ to herself, we have the relation

$$\frac{v_j^{t_{OPT}}}{v_j^{t_j}} = \frac{v_j f(t_{OPT}) + d(t_{OPT})}{v_j f(t_j) + d(t_j)} \tag{13}$$

$$v_j^{t_{OPT}} \leq \frac{(v_j f(t') + d(t'))_{max}}{(v_j f(t') + d(t'))_{min}} v_j^{t_j},\ t' \in [a_i, d_i) \tag{14}$$

- Secondly, we consider all the agents in OPT who is interrupted by j directly or indirectly.

  Lemma 1. *Denote the sum of values of the agents who are allocated at t with V_{-i}. If i is not allocated in time slot t, we have: $V_{-i} \geq v_i^t$.*

Algorithm 1: Item Allocation Algorithm

Input: A time slot $t \in \mathbb{T}$, a set of active agents $\mathbb{N}_a$, a set of ongoing agents $\mathbb{N}_o$, a vector of reported types $\hat{\boldsymbol{\theta}}$, a preemption factor α, a set of temporary winners $\mathbb{W}_{t-1}$ at time slot $t-1$.

Output: A temporary winners $\mathbb{W}_t$ at time slot t and ultima winners $\mathbb{W}$.

foreach $i \in \mathbb{N}$ **do**
  if $i \in \mathbb{N}_o$ **then**
    $\varphi_i \leftarrow (t - t_i(\hat{\boldsymbol{\theta}}))/l_i$;
    $b_i(t) \leftarrow \hat{v}_i(t_i(\hat{\boldsymbol{\theta}})) \times \alpha^{\varphi_i}$;
  if $i \in \mathbb{N}_a \backslash \mathbb{N}_o$ **then**
    $b_i(t) \leftarrow \hat{v}_i(t)$;
$\Gamma \leftarrow \{< b_i(t), m_i >, i \in \mathbb{N}_a\}$;
$\mathbb{W}_t \leftarrow KnapSack(M, \Gamma)$;
foreach $i \in \mathbb{W}_t \backslash \mathbb{W}_{t-1}$ **do**
  $t_i(\hat{\boldsymbol{\theta}}) \leftarrow t$;
foreach $i \in \mathbb{W}_{t-1} \backslash \mathbb{W}_t$ **do**
  $t_i(\hat{\boldsymbol{\theta}}) \leftarrow 0$;
foreach $i \in \mathbb{W}_t$ **do**
  if $t - t_i(\hat{\boldsymbol{\theta}}) + 1 \geq l_i$ **then**
    $\mathbb{W} \leftarrow \mathbb{W} \cup \{i\}$;
return $\mathbb{W}_t, \mathbb{W}$;

Proof. With the property of dynamic programming, we have this property, otherwise i will be allocated. □

Consider agent i who is allocated instances in OPT at t, but she is not allocated in our algorithm. Then we charge the value of i to the winners in proportion with their values: $\frac{v_j^t}{V_{-i}} v_i^t (\leq v_j^t$ by lemma 1). The interruption is divided into two cases:

(i) If i is directly interrupted by j. We have $0 < t - t_j < l_j$, the value charged to j by i is at most $v_j^t = f^{\frac{t-tj}{l_j}} v_j^{t_j}$, at the slot t, there are at most $\frac{W}{w_{min}}$ such jobs interrupted. Thus the value charged to j is at most $\frac{W}{w_{min}} f^{\frac{t-tj}{l_j}} v_j^{t_j}$

(ii) If i interrupted by a chain of agents and the final one is agent j. We have $t - t_j < 0$. Assume agent i is interrupted by agent i_2 after $t_2 - t$ slots, and then agent i_2 is interrupted by agent i_3 after $t_3 - t_2$ slots and so on. Then, we have the value charged to agent j by i is at most $f^{\frac{t-t_j}{l_{max}}} v_j^{t_j}$.

To sum up, the total charged to j is

$$v_j^{t_j}\{\frac{(v_j^{a_j}f(t')+d(t'))_{max}}{(v_j^{a_j}f(t')+d(t'))_{min}} + \frac{W}{w_{min}}(\sum_{t=-\infty}^{t_j} f^{\frac{t-t_j}{l_{max}}} + \sum_{t=t_j+1}^{t_j+l_j} f^{\frac{t-t_j}{l_j}})\}$$

$$\leq v_j^{t_j}\{\frac{(v_j^{a_j}f(t')+d(t'))_{max}}{(v_j^{a_j}f(t')+d(t'))_{min}} + \frac{W}{w_{min}}\sum_{\Delta=-\infty}^{l_{max}} f^{\frac{\Delta}{l_{max}}}\}$$

$$= v_j^{t_j}\{\frac{(v_j^{a_j}f(t')+d(t'))_{max}}{(v_j^{a_j}f(t')+d(t'))_{min}} + \frac{W}{w_{min}}\frac{f}{1-f^{-\frac{1}{l_{max}}}}\}$$

□

Within simple mathematical calculations, we get the optimal preemption factor $f = (1 + \frac{1}{l_{max}})^{l_{max}}$

W	server capacity(assuming constant)
w_{min}	minimum instances required by one job
l_{max}	maximum length of one job
θ_i	agent i's type($= (a_i, d_i, l_i, v_i, w_i)$)
a_i	agent i's true arrival time
d_i	agent i's true departure time
l_i	job length(required slots) of agent i
v_i^t	agent i's value at time t
v_i	agent i's initial value
b_i^t	agent i's proposed valuse(bid) at time t
b_i	agent i's initial bid
w_i	instances required by job of agent i
f	preemption factor($f > 1$)

Table 1: Notations

Before the allocation of instances at each slot, we calculate the virtual bid of agents by multiplying a preemption factor. When a new agent proposes a much higher bid in need of emergency use of the instances but all the instances are occupied by currently running jobs. The auctioneer can choose to interrupt running jobs without charging the agents for partial usage of the instances, or the auctioneer can reject the request of the new agent. Then we adjust the ongoing winners to a virtue bid higher than their original bid to give them priority of being allocated(as shown in line 5 of the pseudo code). After, we allocate the instances with dynamic programming(line 11) and update the allocation states(lines 14∼.)

4.3 Payment Calculation

Traditionally in online auction model, the pricing methods used is taking the minimum of the critical prices of one interval. However, that does not fit in our special setting where agents' value is time varying. Firstly we calculate the critical price for a single slot using the property of dynamic programming algorithm. Besides, we add one new time dimension to the classic payment determination Myerson lemma to derive the payment.

Critical Price Calculation

Firstly, it's necessary that we calculate the minimum bid required for i to win the single slot t: q_i^t. During the KnapSack ([Martello and Toth, 1990]) process of the allocation algorithm, we use 2-dimensional array $F[k][w]$ to denote the maximum value the auctioneer can get from the first k agents with w instances. Suppose i is the last agent of the all the agents to be considered (as the sequence in the dynamic programming algorithm does not matter), we have:

$$F[n][w] = max\{F[n-1][W-w_i]+v_i, F[n-1][W]\} \quad (15)$$

When the function max takes the former term, agent i wins. Otherwise, agent i loses. Thus, by the definition of critical price, we have:

$$q_i^t = max\{F[n-1][W] - F[n-1][W-w_i],\ 0\} \quad (16)$$

Then, we can calculate P_i^t, the critical price for agent i to win l_i contiguous slots (to finish her jot) starting from slot $t(t \in [a_i, d_i - l_i - 1])$:

$$P_i^t = \max_{t' \in [t, t+l_i-1]} \frac{q_i^{t'}}{f^{\frac{t'-t}{l_i}}} \quad (17)$$

Payment Derivation

Lemma 2. ***(Extended Myerson)*** *In the online auction setting where agents' values vary with time:*

(a) An allocation rule $\boldsymbol{x}$ *is implementable if and only if it is monotone.*

(b) Monotonicity of $\boldsymbol{x}$ *implies there is a unique payment rule for interval of one agent such that the mechanism* $(\boldsymbol{x}, \boldsymbol{p})$ *is DSIC.*

Proof. We will show the proof of the unique payment by giving derivation of the payment using the monotonicity property. □

The allocation **x** of our setting is a 0-1 monotone curve. We cleverly invoke the stringent DSIC constraint to narrow down the candidate of **p** to a single one.

Use $t(z)$ to denote the earliest time slot that agent i will win when her initial bid is z. Consider $0 \leq y < z$, suppose agent i has private valuation z and submit the false bid y, we have:

$$[f(t(z))z + d(t(z))]x(z) - p(z) \geq [f(t(y))z + d(t(y))]x(y) - p(y) \tag{18}$$

The left side of (18) is the utility of bidding z, the right side of (9) is the utility of bidding y.

Similarly, suppose agent i has private valuation y and submit the false bid z, we have:

$$[f(t(y))y + d(t(y))]x(y) - p(y) \geq [f(t(z))y + d(t(z))]x(z) - p(z) \tag{19}$$

The left side of (19) is the utility of bidding y, the right side of (19) is the utility of bidding z.

Use the symbol $\circ$ to denote the compound function and oving $p(z)$ and $p(y)$ to the same side of the inequalities, we have the restrictions:

$$\begin{aligned} &[F \circ t(z)y + D \circ t(z)]x(z) - [F \circ t(y)y + D \circ t(y)]x(y) \\ &\leq p(z) - p(y) \\ \leq &[F \circ t(z)z + D \circ t(z)]x(z) - [F \circ t(y)z + D \circ t(y)]x(y) \end{aligned}$$

Denote the function $F \circ t * x$ as function R, $D \circ t * x$ as function S, we have:

$$\begin{aligned} &y[r(z) - r(y)] + S(z) - S(y) \\ &\leq p(z) - p(y) \\ &\leq z[r(z) - r(y)] + S(z) - S(y) \end{aligned} \tag{20}$$

Take the limit: $\lim_{y \to z}(11)$, with the restriction of two sides, we get:

$$jump\, in\, p\, at\, z = z{*}jump\, in\, R\, at\, z{+}jump\, in\, S\, at\, z \tag{21}$$

Naturally, we assume $p(0) = 0$. We can then get the intermediate form of the payment for a certain candidate i,

$$p_i(b_i, b_{-i}) = \sum_{k=1}^{l} z_k * jump\ in\ R_i\ at z_k + jump\ in\ S_i\ at\ z_k \tag{22}$$

where $R_i = F_i \circ t_i * x_i$, $S_i = D_i \circ t_i * x_i$. And $z_1, z_2, ..., z_l$ are the breakpoints of the function $t_i(b)$ in the range $[0, b_i]$ (since F_i and D_i are continue.)

The determination of breakpoints of the function $t_i(b)$ is not hard. It relies on the critical prices of i over the available time period:$\{P_i^t | t \in [a_i, d_i)\}$ and the variation function of i's bid over time. Intuitively, we increase the initial bid b_i and observe the critical points where the curve of function of i's bid over time intersects the set $\{P_i^t | t \in [a_i, d_i]\}$, as illustrated in the example below:

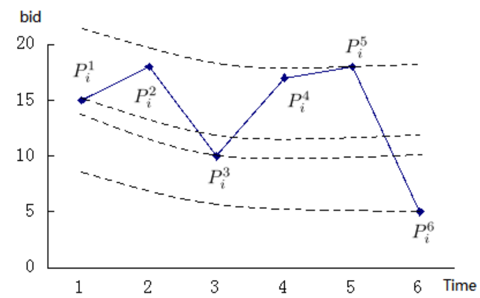

Figure 1: Intuitive example of finding the breakpoints of $t_i(b)$

Here is an intuitive example,

In the example, $F_i(t)$ is in the exponential form: $F_i(t) = \eta^(t - a_i)(0 < \eta < 1)$. In fact, it's not required that b_i^t decrease with time. Here we use the example only for simplicity of demonstration. We observe that $z_1 = \frac{P_i^6 - D_i(6)}{F_i(6)}$, $z_2 = \frac{P_i^3 - D_i(3)}{F_i(3)}$ and $z_3 = \frac{P_i^1 - D_i(1)}{F_i(1)}$ are the breakpoints. And P_i^5 does not generate a breakpoint because for each initial bid b, $t_i(b)$ corresponds to the earliest time slot (i.e., time slot 1). The jumps and breakpoints $z_k (k \in \{1, 2, 3\})$ are marked in Fig.2.

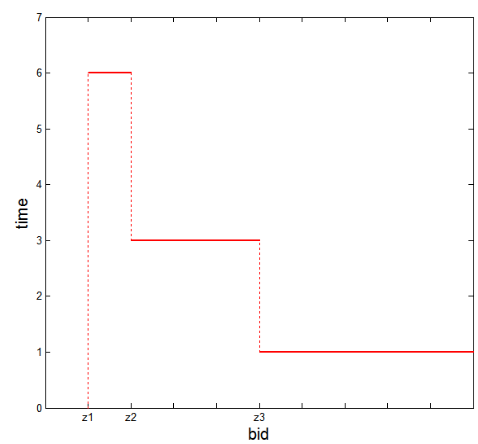

Figure 2: Function $t_i(b)$ with the breakpoints and jumps marked (example)

As for the common case, we give a generalized algorithm to determine the breakpoints of function $t_i(b)$ and the corresponding piece wise constant function. Firstly, we calculate the minimum initial bid for i to win l_i slots from slot $t (t \in [a_i, d_i))$. Yet, not all the candidate initial bid are real breakpoints, in the $while$ loop, we check if the current candidate is invalidated by previous validate ones (i.e., finding a non-decreasing subsequence).

After invoking algorithm 2, we get the function $t_i(b)$ which is illustrated by Fig. 3.

Finally, following equation (22), with the initial bid b_i given, we can get the corresponding payment:

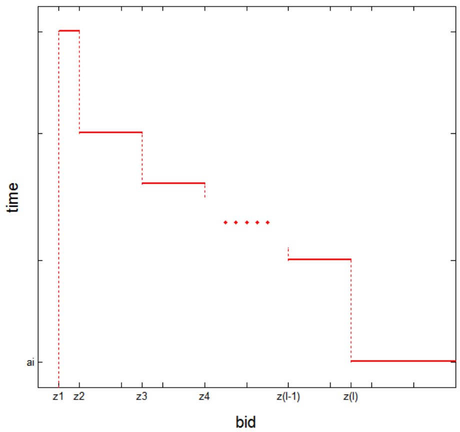

Figure 3: Function $t_i(b)$ with the breakpoints marked

$$p_i(b_i, b_{-i}) = z_1 F_i \circ t_i(1) + D_i \circ t_i(1) \\ + \sum_{k=2}^{k=p} \{ z_k | F_i \circ t_i(z_k) - F_i \circ t_i(z_{k-1}) + | D_i \circ t_i(z_k) - D_i \circ t_i(z_{k-1}) | \} \tag{23}$$

where $z_1 \le z_2 \le ... \le z_p \le b_i \le z_{p+1} \le ... \le z_l$

Theorem 3. *In the case where agents have decreasing values, our mechanism can resist agent i from improving her utility by setting its arrival time $a_i^{'} > a_i$ or departure time $d_i^{'} < d_i$.*

Proof. Use $u_i^{'}$ and u_i be i's utilities when announcing the false arrival time and departure time and when announcing the true type respectively. Let $p_i^{'}$ and p_i be the i's payment in each case. If i's job gets done in both cases, according to the payment formula (23), we have $p_i^{'} \ge p_i$. Then suppose agent i honestly reports her value and gets allocated at slot t and t' respectively, we have, $u_i = v_i^t - p_i \le v_i^{t'} - p_i^{'} = u_i^{'}$. □

5 Numerical Results

We implement our design in C++, agents' arrivals are assumed to be poison process. We show the change of social welfare, average winning delay, average winning loss with the number of agents as well as the preemption factor. To get the result of the off-line case, we set preemption factor to infinity (i.e., no preemption exists.) We take the average of 1000 runs to get the results.

The evaluation results on social welfare are shown in Figure 4. With the agent number increases, the social welfare increases. We observe that when preemption factor is set at 3, the performance is the best, only within an around 2 competitive ratio with the off-line optimal solution. When the factor is reduced to 0.5 (i.e., fierce preemption) or increased to 10000 (i.e., nearly no preemption) the social welfare is worse.

Figure 5 presents the evaluation results of revenue. Same with social welfare, we see that setting preemption factor at 3 makes the performance better. When the factor is reduced to 0.5 (i.e., fierce preemption) or increased to 10000 (i.e., nearly

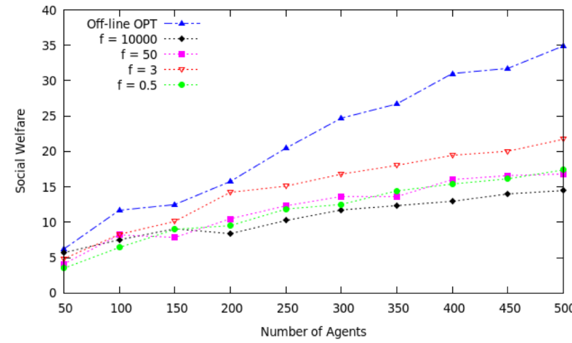

Figure 4: Comparison on Social Welfare

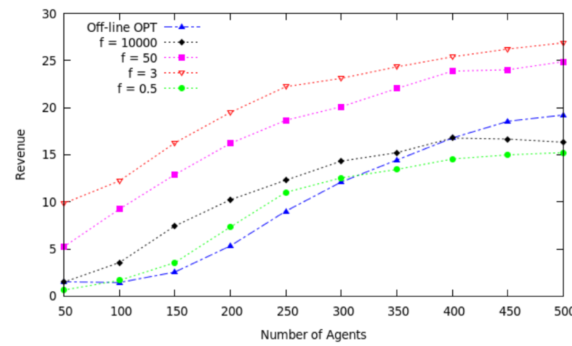

Figure 5: Comparison on Revenue

no preemption) the revenue is decreased, whose performance are worse than that of the off-line optimal solution when number of agents is more than 350. The reason is mainly that when number of agent increase, the preemption happens more frequently and fewer initially allocated agents can keep possessing the resources until its job ends. Consequently, we arrive at the conclusion that introducing the preemption to an appropriate extent can effectively increase revenue of the auctioneer.

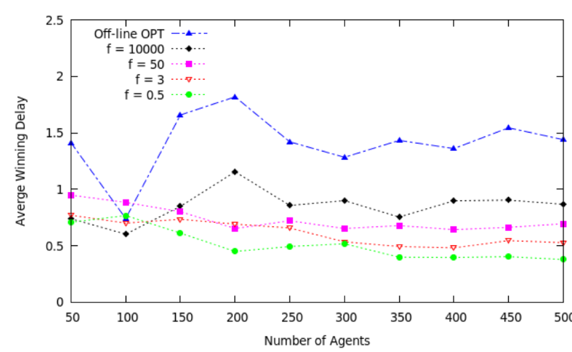

Figure 6: Comparison on Average Wining Delay

Figure 6 shows the evaluation results on average winning delay. Figure 7 shows the evaluation results on average valuation loss, which are almost proportional to the results of winning delay. We can observe that our mechanism with various

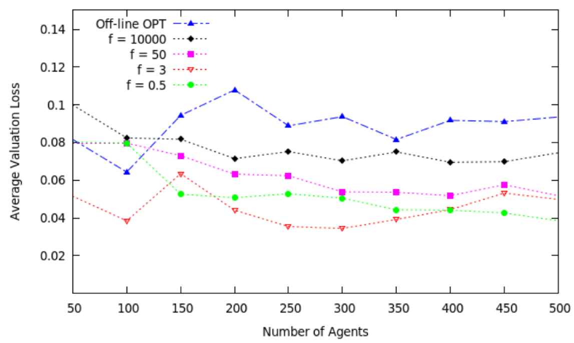

Figure 7: Comparison on Average Valuation Loss

preemption factor all achieve lower average winning delay than off-line optimal solution. In our solution, as preemption is allowed to happen, the winners can get allocated sooner as long as their initial bid is high enough. Besides, when preemption exists, the higher the factor (i.e., the more we restrict preemption), the larger the average winning delay is. It is compatible with our expectation. As we have set the number of slots to be finite which can affect the performance of our design, we find that the results on average winning delay and average valuation loss do not grow monotonously with number of agents. Particularly when the number of agents is around 100, our design achieves the lowest delay.

6 Related Work

In the literature of multiagent system and economics, online mechanism design is an important topic. There have been extensive research in this field. [Lavi and Nisan, 2000] initiated the study of online auction in the domain of computer science. [Parkes and Singh, 2003], [Gershkov and Moldovanu, 2010] and [Nisan *et al.*, 2007] developed online variants of Vickrey-Clarke-Groves (VCG) mechanisms. Their focus is on Bayesian-Nash incentive compatibility and on a model of future supply, as well as future availability to tackle with the problem of lack of knowledge of future arrivals (e.g., [Parkes and Singh, 2003] used an MDP-type framework for predicting future arrivals). As for the early applications of the online mechanism, [Friedman and Parkes, 2003] suggested the allocation of Wi-Fi bandwidth at Starbucks and [Porter, 2004] proposed the model for scheduling of jobs on a server. However, none of the mechanisms mentioned above considered time varying values of agents as well as introducing preemption to improve the auction efficiency.

Later in 2005, [Hajiaghayi, 2005] studied the problem of online auction of a single and re-usable item over a finite time interval in the model-free settings. They derived the lower bound competitive ratios for these designs. [Xu and Li, 2009] proposed online auction designs with spectrum reuse and preemption which did not solve the problem of cheating on arrival and departure time. [Zhou and Zheng, 2009] designed truthful auctions with resource reuse by using periodic auctions. [Deek *et al.*, 2011] integrated online resource allocation and pricing with flexible preemption and guaranteed truthfulness in arrival/deadline. However, [Deek *et al.*, 2011] did not consider time varying and assumed that the demand can only be a single channel.

Furthermore, [Wu *et al.*, 2014] considered online auctions with discounting values which assumes single unit demand and single unit length of agents' job. However, the assumptions are rarely met in reality. Besides, its pricing strategy is complicated and does not apply to the general cases (e.g., time varying values, agents' different time varying factor functions.) [Gerding *et al.*, 2011] proposed an interesting online auction protocol for electric vehicles charging, where owners have non-increasing marginal valuations for each subsequent unit of electricity. Strictly speaking, although the value vector is non-increasing in their settings, one element of the vector, which maps to value for one item in our setting, dose not vary with time.

7 Conclusions

We consider the online auctions with multi-units instances where agents have time varying values. We propose a semi strategy-proof mechanism. Our design resists bid-cheating in the general case and resist misreport of arrival and departure times in the special decreasing value case. In addition, we use preemption factor to adjust agents' values to improve auction efficiency.

There remain some problems to be worked on in the future:

- Multiple non-identical items. In this work, we assume the instances to be allocated are identical. But assuming non-identical instances seems closer to reality. Still use online resource allocation in cloud computing as an example, one may require multiple kinds of resources (e.g., CPU and RAM) simultaneously.
- Resisting misreports in the general case. Another direction is to consider the methods to resist agents' gain in utility when reporting later arrival or early departure in the varying values case.

References

[Deek *et al.*, 2011] Lara Deek, Xia Zhou, Kevin Almeroth, and Haitao Zheng. To preempt or not: Tackling bid and time-based cheating in online spectrum auctions. In *INFOCOM*, 2011.

[Friedman and Parkes, 2003] Eric J Friedman and David C Parkes. Pricing wifi at starbucks: issues in online mechanism design. In *Proceedings of the 4th ACM conference on Electronic commerce*, 2003.

[Gerding *et al.*, 2011] Enrico H Gerding, Valentin Robu, Sebastian Stein, David C Parkes, Alex Rogers, and Nicholas R Jennings. Online mechanism design for electric vehicle charging. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 2011.

[Gershkov and Moldovanu, 2010] Alex Gershkov and Benny Moldovanu. Efficient sequential assignment with incomplete information. *Games and Economic Behavior*, 68(1):144 – 154, 2010.

[Hajiaghayi, 2005] Mohammad T Hajiaghayi. Online auctions with re-usable goods. In *Proceedings of the 6th ACM conference on Electronic commerce*, 2005.

[Lavi and Nisan, 2000] Ron Lavi and Noam Nisan. Competitive analysis of incentive compatible on-line auctions. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 2000.

[Martello and Toth, 1990] Silvano Martello and Paolo Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[Nisan *et al.*, 2007] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge, 2007.

[Parkes and Singh, 2003] David C Parkes and Satinder P Singh. An mdp-based approach to online mechanism design. In *Advances in Neural Information Processing Systems*, 2003.

[Porter, 2004] Ryan Porter. Mechanism design for online real-time scheduling. In *Proceedings of the 5th ACM conference on Electronic commerce*, 2004.

[Shi *et al.*, 2014] Weijie Shi, Linquan Zhang, Chuan Wu, Zongpeng Li, and Francis Lau. An online auction framework for dynamic resource provisioning in cloud computing. In *The 2014 ACM international conference on Measurement and modeling of computer systems*, 2014.

[Wu *et al.*, 2014] Fan Wu, Junming Liu, Zhenzhe Zheng, and Guihai Chen. A strategy-proof online auction with time discounting values. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[Xu and Li, 2009] Ping Xu and Xiang-Yang Li. Online market driven spectrum scheduling and auction. In *ACM Workshop on Cognitive Radio Networks*, 2009.

[Zhou and Zheng, 2009] Xia Zhou and Heather Zheng. Trust: A general framework for truthful double spectrum auctions. In *INFOCOM*, 2009.