Truthful Online Scheduling with Commitments

EC 2015

Problem Glance

Online scheduling

Allowing preemption

Commitments

Problem Glance (cond’t)

Online scheduling

- Jobs arrive in an online manner
- A job includes: (<u>valuation</u>, <u>demand</u>, <u>arrival time</u>, <u>deadline</u>)
- A job holds no value unless it is fully completed
- Need to pay for the completed job

Preemption

- New arrival job with enough high valuation density could suspend the executing job, and then runs itself

Problem Glance (cond't)

Truthfulness

- Users have incentives to reveal their true information for maximizing their utility

Commitment

- A Job will obtain a response whether or not it could be finished before its deadline, even if the scheduling system allows preemption

Detailed Model (Bidding Model)

- Bid: $\tau_j = \langle v_j, D_j, a_j, d_j \rangle$
- Single-minded
- <u>At most one job</u> runs in one server at a specific time

Detailed Model (Other Parameters & Objective)

- (Only discuss the single server case)
- Bids set: $\tau = \{\tau_j : j \in \mathcal{J}\}$ $\tau_j = \langle v_j, D_j, a_j, d_j \rangle$
- Valuation density: $\rho_j = v_j / D_j$
- Density classes: $\boxed{\mathcal{C}_\ell = \{j \mid \rho_j \in [\gamma^\ell, \gamma^{\ell+1})\}}$
- Slackness: $s = \min_{j \in \mathcal{J}} \{ \frac{d_j - a_j}{D_j} \mid \tau_j = \langle v_j, D_j, a_j, d_j \rangle \in \tau \}$

- Objective: maximize the total valuation of fully completed jobs

Truthful Non-committed Scheduling (Algorithm)

ALGORITHM 1: Truthful Non-Committed Algorithm $\mathcal{A}_T$ for a Single Server

$\forall t, \quad J^P(t) = \{ j \in \mathcal{J} \mid j$ partially processed by $\mathcal{A}_T$ at time $t \wedge t \in [a_j, d_j]\}$.

$\quad J^E(t) = \{ j \in \mathcal{J} \mid j$ unallocated by $\mathcal{A}_T$ at time $t \wedge \boxed{t \in [a_j, d_j - \mu D_j]\}.}$

Event: On arrival of job j at time $t = a_j$:

1. call ClassPreemptionRule(t).

Event: On completion of job j at time t:

1. resume execution of job $j' = \arg\max \{\rho_{j'} \mid j' \in J^P(t)\}$.
2. call ClassPreemptionRule(t).
3. delay the output response of j until time d_j.

ClassPreemptionRule (t):

1. $j \leftarrow$ job currently being processed.
2. $j^* \leftarrow \arg\max \{\rho_{j^*} \mid j^* \in J^E(t)\}$.
3. if $(j^* \succ j)$:
   - 3.1. preempt j and run j^*.

Truthful Non-committed Scheduling (Example)

$\tau_j = \langle v_j, D_j, a_j, d_j \rangle \quad \mathcal{C}_\ell = \{j | \rho_j \in [\gamma^\ell, \gamma^{\ell+1})\}$

$\rho_j = v_j / D_j \quad \mu = 3 \quad \gamma = 2$

$\mathcal{C}_0 = [1, 2),\ \mathcal{C}_1 = [2, 4),\ \mathcal{C}_2 = [4, 8)$

$\tau_1 = \langle 6, 2, 2, 20 \rangle,\ \rho_1 = 3,\ \rho_1 \in \mathcal{C}_1,\ d_1 - \mu D_1 = 14$

$\tau_2 = \langle 7, 1, 3, 40 \rangle,\ \rho_2 = 7,\ \rho_2 \in \mathcal{C}_2,\ d_2 - \mu D_2 = 37$

$\tau_3 = \langle 1, 1, 3, 7 \rangle,\ \rho_3 = 1,\ \rho_3 \in \mathcal{C}_0,\ d_3 - \mu D_3 = 4$

High class job preempts lower class job

Unallocated job will be dropped at time d_j-μD_j

Truthful Non-committed Scheduling (Truthful & Competitive Ratio)

- Monotone → Truthfulness [Hajiaghayi et al. 2005]

$$\boxed{\tau_j \succ \tau_{j'}, \text{ if } v_j \geq v_{j'}, D_j \leq D_{j'}, a_j \geq a_{j'}, d_j \leq d_{j'}}$$

$$\text{for any } \tau_j \succ \tau_{j'}, \ \mathcal{A}_j(\tau_j, \tau_{-j}) \geq \mathcal{A}_j(\tau_{j'}, \tau_{-j'})$$

- Competitive-ratio (by Dual Fitting [Lucier et al. 2013]):

$$cr_{\mathcal{A}_T}(s) = 2 + \Theta\left(\frac{1}{\sqrt[3]{s}-1}\right) + \Theta\left(\frac{1}{(\sqrt[3]{s}-1)^3}\right), s > 1$$

$$\gamma = \frac{\sqrt{\mu}}{\sqrt{\mu}-1}$$

$$\mu \approx s^{2/3}$$

Next, introduce the committed scheduling

Committed Scheduling (Concept of Commitment)

- Completion guarantee
  - At time $d_j - \omega(d_j - a_j)$, the user could know whether or not its job could be completed
  - Named ω-responsive

Committed Scheduling (Algorithm Rationale)

- Two key components:
  - simulator (virtual)
  - server (actual)

- First construct virtual jobs by arrival jobs, and then run them on the simulator (using A_{T}, i.e., the non-committed scheduling)

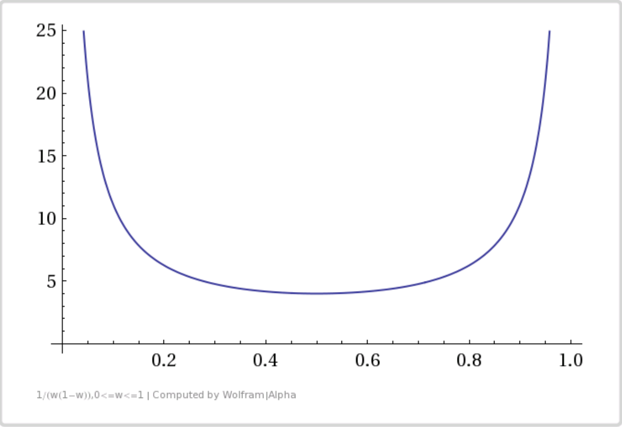
- If a virtual job could be fully completed on the simulator, and then it will be executed in the server

Committed Scheduling (Detailed Parameters)

$\omega \in (0,1)$

- Virtual demand (increased): $D_j^{(v)} = D_j/\omega$
- Virtual deadline (antedated): $d_j^{(v)} = d_j - \omega(d_j - a_j)$
- Virtual job: $\tau_j^{(v)} = \left\langle v_j, D_j^{(v)}, a_j, d_j^{(v)} \right\rangle$
- Slackness assumption (with figure)

$$\forall \tau_j^{(v)}: \quad D_j/\omega \leq (1-\omega)sD_j \Rightarrow s \geq \frac{1}{\omega(1-\omega)}$$



Committed Scheduling (Algorithm)

- Simulator: run virtual jobs by non-committed algorithm

$$\tau_j = \langle v_j, D_j, a_j, d_j \rangle \Rightarrow \begin{array}{l} d_j^{(v)} = d_j - \omega(d_j - a_j) \\ D_j^{(v)} = D_j/\omega \end{array} \Rightarrow \boxed{\tau_j^{(v)} = \left\langle v_j, D_j^{(v)}, a_j, d_j^{(v)} \right\rangle}$$

- Server: Earliest Deadline First (EDF) allocation rule with new admitted jobs
  - An admitted job corresponds with the fully completed virtual job in simulator

$$\tau_j = \langle v_j, D_j, a_j, d_j \rangle$$

$$\tau_j^{(v)} = \left\langle v_j, D_j^{(v)}, a_j, d_j^{(v)} \right\rangle$$

$$\boxed{\tau_j^{(a)} = \left\langle v_j, D_j, d_j^{(v)}, d_j \right\rangle}$$

Committed Scheduling
(How to convert into virtual job & run)

Committed Scheduling (Truthfulness in Public Arrival Time)

- Almost truthful except the arrival time

- Example: misreporting the arrival time

Committed Scheduling (Competitive Ratio)

$$cr_{\mathcal{A}_C}(s) \leq \frac{cr_{\mathcal{A}}\left(s \cdot \omega(1-\omega)\right)}{w(1-\omega)}, \quad s > \frac{1}{\omega(1-\omega)}$$

Advantages

- Responsiveness (commitment)
- No early processing (execute after being admitted)

Thanks

Backup – Relaxed Primal Problem (for Dual Fitting)

$$\max \quad \sum_{j \in \mathcal{J}} \sum_{i=1}^{C} \int_{a_j}^{d_j} \rho_j y_j^i(t) dt$$

$$\sum_{i=1}^{C} \int_{a_j}^{d_j} y_j^i(t) dt \ \leq \ D_j \qquad \forall j$$

$$\sum_{j : t \in [a_j, d_j]} y_j^i(t) \ \leq \ 1 \qquad \forall i, t$$

$$\sum_{i=1}^{C} y_j^i(t) - \frac{1}{D_j} \cdot \sum_{i=1}^{C} \int_{a_j}^{d_j} y_j^i(t) dt \ \leq \ 0 \qquad \forall j, t \in [a_j, d_j]$$

$$y_j^i(t) \geq 0 \qquad \forall j, i, t \in [a_j, d_j]$$

Backup – Relaxed Dual Problem (for Dual Fitting)

$$
\begin{aligned}
\textbf{min} \quad & \sum_{j \in \mathcal{J}} D_j \alpha_j + \sum_{i=1}^{C} \int_0^\infty \beta_i(t) dt \\
\textbf{s.t.} \quad & \alpha_j + \beta_i(t) + \pi_j(t) - \frac{1}{D_j} \int_{a_j}^{d_j} \pi_j(t') dt' \ \geq \ \rho_j && \forall j \in \mathcal{J},\, i,\, t \in [a_j, d_j] \\
& \alpha_j,\, \beta_i(t),\, \pi_j(t) \ \geq \ 0 && \forall j \in \mathcal{J},\, i,\, t \in [a_j, d_j]
\end{aligned}
$$

Backup – Single Compared with Multiple (Competitive Ratio A_{T})

- Single-server:

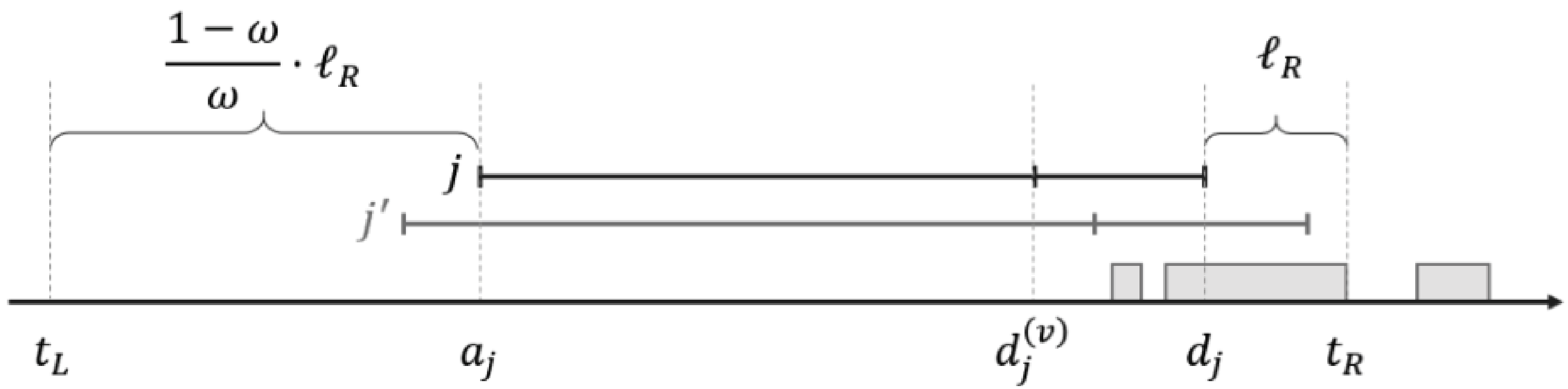
$$cr_{\mathcal{A}_T}(s) = 2 + \Theta\left(\frac{1}{\sqrt[3]{s}-1}\right) + \Theta\left(\frac{1}{(\sqrt[3]{s}-1)^3}\right), s > 1$$

- Multiple-servers:

$$cr_{\mathcal{A}_T}(s) = 2 + \Theta\left(\frac{1}{\sqrt[3]{s}-1}\right) + \Theta\left(\frac{1}{(\sqrt[3]{s}-1)^3}\right), s > 1$$

- The constants hidden inside Θ are slightly larger for the multiple-server case

Backup – Illustration of Commitment Proof



Backup – Fully Truthfulness

- Multiple simulators

- Competitive ratio

$$\mathrm{cr}_{\mathcal{A}_{TC}}(s) = c_0 + \Theta\left(\frac{1}{\sqrt[3]{s/s_0} - 1}\right) + \Theta\left(\frac{1}{(\sqrt[3]{s/s_0} - 1)^3}\right), \quad s > \max\{s_0, 12\beta\},$$

where $c_0 = 187.496$ *and* $s_0 = 279.744$. *For the single server case,* $c_0 = 17$ *and* $s_0 = 24$.