ONLINE COMPANION

Proof of Remark 1

Consider a feasible selection of jobs. It follows from Smith (1956) that, without loss of generality, the sequence given in the Initialization step minimizes the lowest total weighted completion time for that selection. Further, since $q_1 = \cdots = q_T$, scheduling the processed jobs with no inserted idle time is optimal. Finally, since the system values of all feasible selections of jobs are compared within the state transitions, Algorithm GS generates the maximum system value.

Regarding the time complexity, in the recurrence relation, there are $O(n)$ values of i and $O(T)$ values of t. For each i and t, there is a fixed number of values to compare. Therefore, the overall time complexity of Algorithm GS is $O(nT)$. $\square$

Proof of Remark 2

The proof is by reduction from Partition: Given $2m$ elements with integer sizes $a_1, \ldots, a_{2m}$, where $\sum_{i=1}^{2m} a_i = 2A$, does there exist a partition S_1, S_2 of the index set $\{1, \ldots, 2m\}$ such that $\sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i = A$?

Given an instance of Partition, we construct an instance of the scheduling problem where $n = 2m$, $p_i = a_i$ for $i = 1, \ldots, n$, and $T = A$. Then, the scheduling problem has the maximum possible value A if and only if the instance of Partition has a solution. $\square$

Proof of Remark 3

a. In an optimal schedule for any instance I, the total system value is $V_a(I) + V_b(I) \leq 2\max\{V_a(I), V_b(I)\}$, where $V_a(I)$ is the total reserve value of the unused time slots, and $V_b(I)$ is the total value of the processed jobs to their agents. We show that Algorithm GS delivers a schedule for instance I with total system value of at least $\max\{V_a(I), V_b(I)\}$. First, the total system value of a schedule found by Algorithm GS is at least the total reserve value of all the time slots, and thus at least $V_a(I)$. Second, consider an instance I' that is identical to instance I except that in I', the reserve values of all time slots are 0. From the definition of I', we have $V_b(I') \geq V_b(I)$. Since an optimal schedule for I' is among the feasible schedules compared by Algorithm GS when solving instance I, the algorithm finds a solution that has an objective value of at least $V_b(I') \geq V_b(I)$.

b. Suppose that for instance I there exists an optimal schedule without unallocated time slots. Consider the instance I' defined in part a of the proof. From Remark 1, Algorithm GS finds an optimal schedule for instance I'. The maximum system value in the two instances is the same. Hence, Algorithm GS finds an optimal schedule for instance I. $\square$

Proof of Theorem 1

In any feasible schedule, the admitted bids are sequenced in nondecreasing order of $\overline{u}_i$ values. Since the revenues for the facility owner of all possible state transitions within such sequences are compared, Algorithm WD generates the maximum revenue for the facility owner. Regarding the time complexity, the Initialization step requires $O(n \log n)$ time, and the Optimal Solution Value step requires $O(T)$ time. In the recurrence relation, there are $O(n)$ possible values of i and $O(\overline{u}_n)$ possible values of t. For fixed i and t, only a fixed number of comparisons is required. Therefore, the overall time complexity of Algorithm WD is $O(\max\{n \log n, T, n\overline{u}_n\})$. $\square$

Proof of Theorem 2

Consider an instance with n agents, where $p_i = w_i = 1$ and $v_i = p_n + n$, for $i = 1, \ldots, n-1$; $p_n \geq 1$, $w_n = (p_n + n - 2)/(n-1)$, and $v_n = (p_n + n - 2)(p_n + n - 1)/(n-1)$. The facility owner has $T = p_n + n - 1$ time slots. Let $q_t = 0$, for $t = 1, \ldots, T$.

In an optimal schedule σ^*, all jobs are scheduled consecutively and job n is processed in the last position. The total revenue of schedule σ^* is

$$\sum_{i=1}^{n} v_i = (n-1)(p_n + n) + (p_n + n - 2)(p_n + n - 1)/(n-1); \tag{2}$$

and the scheduling cost S_{σ^*} of schedule σ^* is

$$(n-1)n/2 + (p_n + n - 2)(p_n + n - 1)/(n-1). \tag{3}$$

From (2) and (3), the system value V_{σ^*} of schedule σ^* is

$$(n-1)p_n + n^2/2 - n/2. \tag{4}$$

Now define the prices of fixed time blocks such that $\rho_{(p_n,p_n)} = p_n + n - 2$ and $\rho_{(1,p_n+t)} = n - t - 1$, for $t = 1, \ldots, n-1$. The prices of all other fixed time blocks are:

$$\rho_{(p,\overline{u})} = \begin{cases} \rho_{(p_n,p_n)}, & \text{if } \overline{u} \leq p_n \\ \rho_{(p_n,p_n)} + \sum_{t=p_n+1}^{\overline{u}} \rho_{(1,t)}, & \text{if } \overline{u} \geq p_n + 1 \text{ and } \overline{u} - p \leq p_n - 1 \\ \sum_{t=\overline{u}-p+1}^{\overline{u}} \rho_{(1,t)}, & \text{if } \overline{u} \geq p_n + 1 \text{ and } \overline{u} - p \geq p_n. \end{cases}$$

We consider the above prices and a schedule σ that allocates block (p_n, p_n) to job n, and block $(1, p_n + i)$, $i = 1, \ldots, n-1$, to job i. We show that this solution is in equilibrium at the prices ρ.

First, consider job n. In the current solution, job n generates a profit of $v_n - w_n C_n - \rho_{(p_n, p_n)} = (p_n + n - 2)(p_n + n - 1)/(n-1) - (p_n + n - 2)p_n/(n-1) - (p_n + n - 2) = 0$. If job n completes later than p_n but earlier than $2p_n$, then it incurs both a higher price and a higher work in process cost, and thus generates less profit for its agent. Now suppose job n completes at time $C'_n \geq 2p_n$. The price for time block (p_n, C'_n) is $\rho_{(p_n, C'_n)} = p_n(2n - 2C'_n + 3p_n - 3)/2$. Thus, agent n achieves a profit of

$$\begin{aligned}
& v_n - w_n C'_n - \rho_{(p_n, C'_n)} \\
= \; & \frac{(p_n + n - 2)(p_n + n - 1)}{n-1} - \frac{(p_n + n - 2)C'_n}{n-1} - \frac{p_n(2n - 2C'_n + 3p_n - 3)}{2} \\
\leq \; & \frac{(p_n + n - 2)(p_n + n - 1)}{n-1} - \frac{p_n(2n + 3p_n - 3)}{2} + \frac{(n-2)(p_n - 1)(p_n + n - 1)}{n-1} \\
= \; & \frac{-p_n(p_n - 1)}{2} \\
\leq \; & 0, \quad \text{since } p_n \geq 1.
\end{aligned}$$

Therefore, agent n cannot increase its profit by purchasing other time blocks.

Next, consider job i, where $i \in \{1, \ldots, n-1\}$. In the current solution, agent i gains a profit of $v_i - w_i C_i - \rho_{(1, p_n + i)} = (p_n + n) - (p_n + i) - (n - i - 1) = 1$ for its agent. If job i completes at time $C'_i \leq p_n$, then agent i gains a profit of $v_i - w_i C'_i - \rho_{(1, C'_i)} = (p_n + n) - C'_i - (p_n + n - 2) \leq (p_n + n) - 1 - (p_n + n - 2) = 1$. Otherwise, if job i completes at time C'_i with $p_n + 1 \leq C'_i \leq p_n + n - 1$, then agent i gains a profit of $v_i - w_i C'_i - \rho_{(1, C'_i)} = (p_n + n) - C'_i - (n - (C'_i - p_n) - 1) = 1$. Therefore, agent $i \in \{1, \ldots, n-1\}$ cannot increase its profit by purchasing other time blocks. As a result, the current solution is in equilibrium at the prices ρ.

Since all jobs are scheduled, the total revenue of σ is given by (2). Also, the scheduling cost S_σ of schedule σ is

$$(p_n + n - 2)p_n/(n-1) + (n-1)p_n + (n-1)n/2. \tag{5}$$

From (2) and (5), the system value V_σ of schedule σ is

$$p_n + n^2/2 + n/2 - 2. \tag{6}$$

Then, from (3), (4), (5) and (6), if we let $p_n = n^r$, where $1 < r < 2$, n is integer, and $n \to \infty$, we have

$$\frac{S_\sigma}{S_{\sigma^*}} = \frac{2p_n^2 + (2n^2 - 2n - 2)p_n + n^3 - 2n^2 + n}{2p_n^2 + (4n-6)p_n + n^3 - 5n + 4} \to \infty,$$

and

$$\frac{V_{\sigma^*}}{V_\sigma} = \frac{2(n-1)p_n + n^2 - n}{2p_n + n^2 + n - 4} \to \infty. \qquad \square$$

Proof of Theorem 3

a. We first prove that there exists an optimal schedule in which admitted bids are processed in nondecreasing order of u_i values. Suppose that there exists an optimal schedule σ which contains two blocks B_{i_1} and B_{i_2} such that B_{i_1} precedes B_{i_2} but $u_{i_1} > u_{i_2}$. First, move block B_{i_1} to be completed at the starting time of block B_{i_2}, and then move any blocks that are processed between blocks B_{i_1} and B_{i_2} earlier by p_{i_1}. Next, interchange blocks B_{i_1} and B_{i_2}. Thus, we obtain a new schedule, denoted by σ', which is feasible since $u_{i_1} > u_{i_2}$. Under the assumption of nonincreasing reserve values, the total revenue of schedule σ' for the facility owner is no less than that of the schedule σ. By repeating such transformations, we find a new schedule in which all the processed blocks are in nondecreasing order of u_i values and the total revenue of the facility owner is no less than that of schedule σ. The remainder of the proof is similar to that of Theorem 1.

b. Suppose that, for instance I, there exists an optimal schedule without unallocated time slots. Consider an instance I' that is identical to instance I, except that in I' the reserve values of all time slots are 0. From part a of Theorem 3, Algorithm WD$'$ finds an optimal schedule for instance I'. Observe that the maximum revenue in instance I' is the same as in instance I. Also, note that all schedules considered by WD$'$ for instance I' are also considered for instance I, and in each case the revenue for instance I is at least as large. Therefore, Algorithm WD$'$ finds an optimal schedule for instance I.

c. Recall that an instance defines a set of bids by all the agents. Clearly, Algorithm WD$'$ finds a solution with revenue for the facility owners that is at least that of a schedule found by Algorithm WD. In an optimal schedule for any instance I, the total revenue of the facility owner is $V_a + V_b \leq 2\max\{V_a, V_b\}$, where V_a is the total reserve value of the unallocated time slots, and V_b is the total bid price of the allocated blocks. We show that Algorithm WD$'$ delivers a schedule with total revenue of at least $\max\{V_a, V_b\}$, from which the result follows.

First, the total revenue of a schedule found by Algorithm WD$'$ is at least the total reserve value of all the time slots, and thus at least V_a. Second, consider an instance I' that is identical to instance I except that in I', the reserve values of all time slots are 0. Since accepting the same bids as in the optimal schedule to instance I is a feasible schedule, the maximum revenue in instance I' is no less than V_b. From part a of the theorem, Algorithm WD$'$ finds an optimal schedule for instance I'. Further, all schedules compared by WD$'$ for instance I' are also compared for instance I, and in each case the revenue for instance I is at least as large. Therefore, for instance I, Algorithm WD$'$ finds a schedule with total revenue at least V_b.

d. When all time slots have the same reserve value, it is easy to prove that the problem is binary *NP*-hard, by reduction from the knapsack problem. The unary *NP*-hardness proof for the more general case is by reduction from the following problem, which is known to be unary *NP*-complete.

3-Partition: Given $3m$ elements with integer sizes $a_1, \ldots, a_{3m}$, where $\sum_{i=1}^{3m} a_i = my$ and $y/4 < a_i < y/2$ for $i = 1, \ldots, 3m$, does there exist a partition $S_1, \ldots, S_m$ of the index set $\{1, \ldots, 3m\}$ such that $|S_j| = 3$ and $\sum_{i \in S_j} a_i = y$ for $j = 1, \ldots, m$?

Given an instance of 3-Partition, we construct an instance of the winner determination problem, where $T = my + m$ and $n = 3m$. The reserve values are 1 for time slots $y+1, 2y+2, \ldots, my+m$ and 0 for all the other time slots. The n bids are: $B_1 = \langle (a_1, my+m), 1 \rangle, B_2 = \langle (a_2, my+m), 1 \rangle, \ldots, B_n = \langle (a_n, my+m), 1 \rangle$. It is easy to verify that the instance of the winner determination problem has maximum revenue $4m$ if and only if the instance of 3-Partition has a solution. $\square$

Proof of Remark 5

Proof. Given an optimal schedule σ for an instance I with $k \geq 0$ unallocated time slots for an instance of the winner determination problem, consider the following changes.

1. Process all the admitted blocks in nondecreasing order of u_i values from time 0, without inserted idle time. Denote the new schedule by σ'.
2. Assign the reserve values of all the k unallocated time slots in schedule σ to the last k time slots in schedule σ', in nondecreasing order of reserve values. Denote the new schedule by σ^*.

Observe that schedule σ^* provides the same revenue as schedule σ for the facility owner.

Also, since Algorithm UB assigns all the largest reserve values to the unallocated time slots, it finds a schedule with revenue no less than that of schedule σ^*. Therefore, the value generated by Algorithm UB is an upper bound on the maximum revenue for instance I. $\square$

Proof of Corollary 1

The instance described in the first paragraph of the proof of Theorem 2 is also in equilibrium with flexible time blocks as market goods, hence the result follows. $\square$

Proof of Theorem 4

Proof. We show the existence of an equilibrium solution by construction. Consider an ascending auction where the facility owner optimally solves its winner determination problem at each round. The agents employ the straightforward bidding policy where, among all the time blocks that maximize an agent's profit, the agent bids for one that contains the maximum number of time slots. This bidding policy guarantees that, when an allocated time block is reallocated, all the time slots within it are reallocated. Consequently, the ask price of each time block is nondecreasing throughout the auction, and each agent cannot change its bid to increase its profit. Thus, using the constant bid increment function $\varepsilon(\cdot) = \varepsilon \to 0$, a solution at closure is in equilibrium.

The nonuniqueness of equilibrium solutions is established by the following example. There are two agents, where $p_1 = p_2 = 1$, $v_1 = 5$, $v_2 = 3$, $w_1 = 3$ and $w_2 = 1$. The facility owner has two time slots to sell, with $q_1 = q_2 = 0$. Any solution where agent 1 wins time slot 1 paying $1 \leq \rho_1 \leq 2$, and agent 2 wins time slot 2 paying $0 \leq \rho_2 \leq 1$, is in equilibrium. $\square$

Proof of Remark 7

Consider two agents each with a job with unit processing time, revenue 1, and no scheduling cost, which bid for a single time slot with reserve value 0. The auction reaches closure when an agent bids price b with $0 < b < 1$ and $b+\varepsilon(\cdot) \geq 1$. However, in either of the two nonunique equilibrium solutions, one agent wins the time slot by paying exactly 1. $\square$

Proof of Theorem 5

Let F denote a globally optimal solution, where $\mathcal{A}' \subseteq \mathcal{A}$ is the set of agents with allocated time blocks. If $\mathcal{A}' = \emptyset$, then the proof is complete. Otherwise, let x_j denote the total reserve value of the time slots contained in allocated time block j, where $x_{\max} = \max_j \{x_j\}$. We define prices for the three possible types of time blocks. First, let the price of each time block formed entirely from unallocated time slots be the total reserve value of the time slots it contains. Second, let the price of each fully allocated time block i be $\max\{x_{\max}, \max_{i \in \mathcal{A} \setminus \mathcal{A}'}\{v_i - f_i\}\}$, which is at least the total reserve value of the time slots it contains. Third, there can be time blocks containing both unallocated time slots and time slots from allocated time blocks. For each time block of this type, let its price be the total price of the time blocks containing at least one time slot within the time block, plus the total reserve value of the unallocated time slots within the time block. Observe that these prices satisfy the conditions of the theorem. We complete the proof by showing that the prices also support an equilibrium solution.

We first establish two lower bounds on $\min_{i \in \mathcal{A}'}\{v_i - f_i\}$. First, suppose that $v_i - f_i < x_{\max}$, for some $i \in A'$. Starting from solution F, reallocate the time block of agent i to the agent currently allocated time block j, where $x_j = x_{\max}$. This transformation provides a new solution with an increased total system value, which contradicts the optimality of solution F. Second, if $v_j - f_j > \min_{i \in \mathcal{A}'}\{v_i - f_i\}$ for some $j \in \mathcal{A} \setminus \mathcal{A}'$, agent j will be allocated a time block, which is a contradiction. Hence, $\min_{i \in \mathcal{A}'}\{v_i - f_i\} \geq \max_{i \in \mathcal{A} \setminus \mathcal{A}'}\{v_i - f_i\}$. Combining these two results gives $\min_{i \in \mathcal{A}'}\{v_i - f_i\} \geq \max\{x_{\max}, \max_{i \in \mathcal{A} \setminus \mathcal{A}'}\{v_i - f_i\}\}$.

Consider the agents in $\mathcal{A}'$. The total reserve value of each time block entirely containing at least p unallocated time blocks is at least $\max\{x_{\max}, \max_{i \in \mathcal{A} \setminus \mathcal{A}'}\{v_i - f_i\}\}$, otherwise such a time block will be allocated in solution F. Therefore, bidding for such a time block will not increase the profit of any agent in $\mathcal{A}'$. Also, since the prices of all the allocated time blocks are the same, no agent in $\mathcal{A}'$ can increase its profit by purchasing a different allocated time block. Further, since $\min_{i \in \mathcal{A}'}\{v_i - f_i\} \geq x_{\max}$, no agent in $\mathcal{A}'$ can improve its profit by withdrawing an admitted bid. Finally, the other time blocks have the same value for each agent as the allocated ones, but cost more. Therefore, bidding for them cannot increase the profit of an agent in $\mathcal{A}'$.

Next, consider the agents in $\mathcal{A} \setminus \mathcal{A}'$. From the optimality of solution F, an agent cannot increase its profit by purchasing a time block formed entirely from unallocated time slots. Also, an agent cannot increase its profit by bidding for an allocated time block, since its price is at least $\max_{i \in \mathcal{A} \setminus \mathcal{A}'}\{v_i - f_i\}$. Further, for the other time blocks, similar arguments hold as for the agents in $\mathcal{A}'$.

Finally, consider the facility owner. Note that the price of each time block is no less than the total reserve value of time slots within the time block. When the market goods are fixed time blocks, the schedule is fixed and thus the facility owner cannot improve its revenue. When the market goods are flexible time blocks, given solution F, the facility owner can still change its schedule. However, since the scheduling cost of each job is independent of its completion time, an increase in profit for the facility owner also increases the total system value. This contradicts the optimality of solution F. $\square$

Proof of Remark 8

Since the scheduling cost of each processed job is independent of its completion time, the sequence of the processed jobs does not affect the system value. Since the system values of all possible state transitions are compared, Algorithm GOS generates the maximum system value. The analysis of the time complexity follows the proof of Theorem 1. $\square$