A Omitted Proofs in Section 2

A.1 Proof of Lemma 1

Proof The waiting time computation formula indicates that the queue leader is key to the calculation. When the waiting time of job $\sigma_t(j)$ drops by $\Delta w^-_{\sigma_t(j)} = \Delta$ ($\Delta \geq 0$), it is necessary to classify and discuss whether it is the queue leader.

We label the neighboring jobs $\sigma_t(i)$ and $\sigma_t(i+1)$ as j and k, respectively, for simplicity in the reasoning process, where $j < k$ indicating that job j precedes job k in processing. w_j^2 and w_k^2 are the waiting times of jobs j and k, correspondingly, after the change.

Case 1: $w_j > \Delta$**.**

In this case, job j is neither the queue leader initially nor after the reduction. The queue leader remains unchanged, and the updated waiting time for job j is:

$$w_j^2 = w_j - \Delta > 0.$$

The waiting time of job k is updated as follows:

$$w_k^2 = r_j + w_j^2 + p_j - r_k,$$

since the original waiting time for job k is:

$$w_k = r_j + w_j + p_j - r_k,$$

we can get the waiting time reduction for job k:

$$\begin{aligned}\Delta w_k^- &= w_k - w_k^2 \\ &= r_j + w_j + p_j - r_k - (r_j + w_j^2 + p_j - r_k) \\ &= w_j - w_j^2 = \Delta.\end{aligned}$$

Case 2: $0 < w_j \leq \Delta$**.**

In this scenario, we have $w_j^2 = w_j - \Delta \leq 0$, meaning that job j becomes the new queue leader and starts processing immediately.

The new waiting time for job k is:

$$w_k^2 = r_j^o + p_j - r_k,$$

since its original waiting time is:

$$w_k = r_j + w_j + p_j - r_k,$$

the waiting time reduction of job k is:

$$\begin{aligned}\Delta w_k^- &= w_k - w_k^2 \\ &= r_j + w_j + p_j - r_k - (r_j^o + p_j - r_k) \\ &= w_j.\end{aligned}$$

(The superscript o is only an indicator of the queue leader and does not change the values of the release time.)

Case 3: $w_j \leq 0$**.**

In this case, job j is already the queue leader initially, and a further reduction in its waiting time does not affect subsequent jobs. The updated waiting time for job k is:

$$w_k^2 = r_j^o + p_j - r_k,$$

and the original waiting time for job k is:

$$w_k = r_j^o + p_j - r_k,$$

the waiting time reduction for job k can be get:

$$\begin{aligned}\Delta w_k^- &= w_k - w_k^2 \\ &= r_j^o + p_j - r_k - (r_j^o + p_j - r_k) \\ &= 0.\end{aligned}$$

Conclusion: We can conclude from these cases that: If $0 < w_j \leq \Delta$, the decrease in waiting time of job k is w_j; if $\Delta < w_j$, job k's waiting time reduced Δ. It always takes the smaller of the two.

$$\begin{aligned}\Delta w_k^- &= \min(\Delta, w_j) \\ &= \min(\Delta w_j^-, w_j).\end{aligned}$$

If $w_j \leq 0$, then:

$$\Delta w_k^- = 0.$$

We can express it concisely with a mathematical formula as follows:

$$\Delta w_k^- = \begin{cases} 0, & \textit{if } w_{\sigma_t(k)} \leq 0, \\ \min(\Delta w_j^-, w_j), & \textit{if } w_{\sigma_t(k)} > 0. \end{cases}$$

This conclusion can be generalized to the waiting time changes of all subsequent jobs, finally obtaining a standard recursive formula, when $w_{\sigma_t(j)}$ is decreased:

$$\Delta w_{\sigma_t(k+1)}^- = \begin{cases} 0, & \textit{if } w_{\sigma_t(k)} \leq 0, \\ \min(\Delta w_{\sigma_t(k)}^-, w_{\sigma_t(k)}), & \textit{if } w_{\sigma_t(k)} > 0. \end{cases}$$

where the initial condition is $w_{\sigma_t(j)} = \Delta$ and $k \in [j, n]$. □

A.2 Proof of Lemma 2

Proof Similar to the demonstration in Lemma 1, the adjacent jobs $\sigma_t(i)$ and $\sigma_t(i+1)$ are denoted as j and k, respectively, with $j < k$ meaning that job j precedes job k in processing. w_j^2 and w_k^2 are the revised waiting times for jobs j and k, correspondingly, following the adjustment.

Case 1: $w_j + \Delta \leq 0$**.** Job j is the queue leader initially. Even after its waiting time increases by Δ $w_j^2 = w_j + \Delta \leq 0$, it remains the queue leader.

Since job j still leads the queue, the waiting time of job k remains unchanged:

$$\begin{aligned}w_k^2 &= r_j^o + p_j - r_k, \\ w_k &= r_j^o + p_j - r_k.\end{aligned}$$

Hence, the increase in waiting time for job k is:

$$\Delta w_k^+ = w_k^2 - w_k = 0.$$

Case 2: $w_j \leq 0$ and $0 < w_j + \Delta$. Job j is the queue leader at the beginning. It is no longer the head when its waiting time is increased Δ, and becomes a member of the continuous queue.

$$w_j^2 = w_j + \Delta > 0.$$

The waiting time of job k updates as:

$$w_k^2 = r_j + w_j^2 + p_j - r_k,$$

given the original waiting time:

$$w_k = r_j^o + p_j - r_k,$$

the waiting time increase for job k is:

$$\begin{aligned}
\Delta w_k^+ &= w_k^2 - w_k \\
&= r_j + w_j^2 + p_j - r_k - (r_j^o + p_j - r_k) \\
&= w_j^2 \\
&= w_j + \Delta.
\end{aligned}$$

Case 3: $w_j > 0$. Job j is not the queue leader, increasing its waiting time does not change its queue status$w_j^2 = w_j + \Delta > 0$.

The updated waiting time of job k is:

$$w_k^2 = r_j + w_j^2 + p_j - r_k.$$

Originally:

$$w_k = r_j + w_j + p_j - r_k.$$

Last, the waiting time increase for job k is:

$$\begin{aligned}
\Delta w_k^+ &= w_k^2 - w_k \\
&= r_j + w_j^2 + p_j - r_k - (r_j + w_j + p_j - r_k) \\
&= w_j^2 - w_j \\
&= \Delta.
\end{aligned}$$

Conclusion: From these cases, we conclude: If $w_j \leq 0$ and $w_j + \Delta \leq 0$, then we get $\Delta w_k^+ = 0$. If $w_j \leq 0$ and $w_j + \Delta > 0$, it will be $\Delta w_k^+ = w_j + \Delta$. Δw_k^+ is always the bigger one of 0 and $w_j + \Delta$.

$$\begin{aligned}
\Delta w_k^+ &= \max(0, w_j + \Delta) \\
&= \max(0, w_j + \Delta w_j^+).
\end{aligned}$$

When $w_j > 0$, it seems that only one result exists:

$$\Delta w_k^+ = \Delta = \Delta w_j^+.$$

Finally, we get the concisely mathematical formula:

$$\Delta w_k^+ = \begin{cases} \Delta w_j^+, & \textit{if } w_k > 0, \\ \max(0, \Delta w_j^+ + w_j), & \textit{if } w_k \leq 0. \end{cases}$$

This conclusion can be further broadened to the waiting time changes of all subsequent jobs, finally, a general recursive formula is obtained, when $w_{\sigma_t(j)}$ is increased:

$$\Delta w_{\sigma_t(k+1)}^+ = \begin{cases} \Delta w_{\sigma_t(k)}^+, & \textit{if } w_{\sigma_t(k)} > 0, \\ \max(0, \Delta w_{\sigma_t(k)}^+ + w_{\sigma_t(k)}), & \textit{if } w_{\sigma_t(k)} \leq 0. \end{cases}$$

where the initial condition is $\Delta w_{\sigma_t(j)}^+ = \Delta$ and $k \in [j, n]$. □

A.3 Proof of Proposition 1

Proof Consider an arbitrary schedule σ_t and any three consecutive jobs $\sigma_t(i)$, $\sigma_t(i+1)$, and $\sigma_t(i+2)$, with $i \in 1, \dots, n-2$. Without loss of generality, denote these three jobs by 4, 5, and 6, respectively, that is, let $\sigma_t(i) = 4$, $\sigma_t(i+1) = 5$, and $\sigma_t(i+2) = 6$. This relabeling is adopted purely for notational simplicity and has no effect on the validity of the analysis. We focus on the situation where the waiting time of job 4 is reduced by $\Delta w_4^- = \Delta$, with $\Delta \geq 0$.

Situation 1: $w_4 \leq 0$.

Job 4 is the queue leader at the beginning, and with its waiting time decreased Δ, $w_4^2 = w_4 - \Delta \leq 0$, leading to the impact on the total waiting time.

$$\begin{aligned}\Delta w &= \max(0, w_4) - \max(0, w_4^2) \\ &= 0 - 0 = 0.\end{aligned}$$

We can get the values of Δw_5^- according to the recurrence relation$\Delta w_5^- = 0$, so the changes of the total waiting time equal with Δw_5^-, $\Delta w = \Delta w_5^-$.

Situation 2:$w_4 > 0$ and $w_4 - \Delta > 0$. Job 4 is the member of the continuous queue initially and remains so after decreasing its waiting time by Δ. The updated waiting time for job 4 is $w_4^2 = w_4 - \Delta > 0$.

The change in the total waiting time is:

$$\begin{aligned}\Delta w &= \max(0, w_4) - \max(0, w_4^2) \\ &= w_4 - (w_4 - \Delta) \\ &= \Delta.\end{aligned}$$

$\Delta w_5^- = \min(\Delta, w_4) = \Delta$ according to the recurrence formulation. Obviously, $\Delta w = \Delta w_5^-$.

Situation 3: $w_4 > 0$ and $w_4 - \Delta \leq 0$.

In this situation, job 4 transitions to become the new queue leader from the ones of the continuous queue, with $w_4^2 = w_4 - \Delta \leq 0$. Δw_5^- is calculated as:

$$\Delta w_5^- = \min(\Delta, w_4) = w_4.$$

The total waiting time change is:

$$\begin{aligned}\Delta w &= \max(0, w_4) - \max(0, w_4^2) \\ &= w_4 - 0 = w_4.\end{aligned}$$

It seems clear that $\Delta w = \Delta w_5^-$.

Conclusion: The reduction in job 4's waiting time, lowering the total waiting time by Δw, precisely equals its backward transmission value Δw_5^-, thereby proving the proposition. This conclusion holds for all subsequent jobs, following the same proof process.

Similarly, When the waiting time of job 4 increases $\Delta w_4^+ = \Delta$ $(\Delta \geq 0)$, using the same method of analysis can verify that the total waiting time increase, Δw, is the same as Δw_5^+. □

A.4 Proof of Theorem 1

Proof We prove Theorem 1 in two steps. First, we establish the claimed propagation pattern by examining a one-position forward move. We then consider a two-position forward move and show that further forward moves only add processing-time and waiting-time information, without creating any new structural components. Therefore, the pattern admits a direct extension to arbitrarily long forward moves.

Consider an arbitrary schedule σ_t and any four consecutive jobs $\sigma_t(i)$, $\sigma_t(i+1)$, $\sigma_t(i+2)$, and $\sigma_t(i+3)$, where $i \in 1, \dots, n-3$. We begin with the simplest case, in which job $\sigma_t(i)$

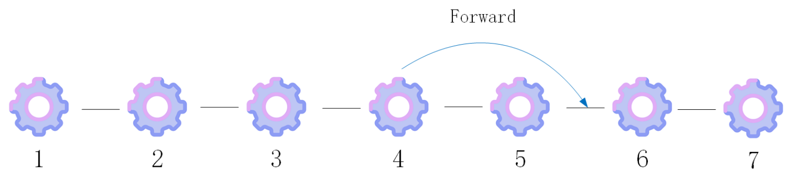

Fig. 1 Job 4 moves to the position between jobs 5 and 6.

is moved to the position immediately after $\sigma_t(i+1)$, as illustrated in Figure 1. For ease of exposition and without loss of generality, let $\sigma_t(i) = 4$, $\sigma_t(i+1) = 5$, $\sigma_t(i+2) = 6$, and $\sigma_t(i+3) = 7$. This relabeling is introduced solely for notational convenience and does not affect the subsequent derivation.

Before starting the derivation, we need to define certain notations and provide proof of the impact of removing job 4 on the waiting time of job 5.

First, we prove that the removal of job 4 will decrease the waiting time of job 5 by $\Delta w_5 = p_4 - \min(0, w_4)$, and the proof processes as follows:

Suppose r_0^o and p_0^o are the release time of the previous queue leader and the cumulative processing time up to job 4, respectively.

- When $w_4 > 0$, the original waiting time of job 5 is $w_5 = r_0^o + p_0^o + p_4 - r_5$, after we remove job 4, the waiting time of job 5 is updated as $w_5^2 = r_0^o + p_0^o - r_5$, the differ between the two is $\Delta w_5 = w_5 - w_5^2 = p_4$.
- When $w_4 \le 0$, the waiting time of job 5 starts with $w_5 = r_0^o + p_0^o - w_4 + p_4 - r_5$, after the removal of job 4, we get the new waiting time of job 5 $w_5^2 = r_0^o + p_0^o - r_5$, the differ of the two is $\Delta w_5 = w_5 - w_5^2 = p_4 - w_4$.

We combine the two cases to reach the conclusion:$\Delta w_5 = p_4 - \min(0, w_4)$.

Second, we record the effect of job i on the total waiting time as Δ_i. For example, if $w_5 > 0, w_5^2 = w_5 - \Delta w_5 \le 0$, the total waiting time is reduced $\Delta_5 = -w_5$, instead of Δw_5. Likewise, when $w_5 > 0, w_5^2 = w_5 - \Delta w_5 > 0$, the total waiting time is decreased $\Delta_5 = -\Delta w_5$. So, the formula of Δ_5 is that $\Delta_5 = \max(0, w_5^2) - \max(0, w_5)$. Actually, $\Delta_i = -\Delta w_{i+1}$ as Proposition 2 suggests. We use $\Delta_5 = -\Delta w_6^-$ to calculate the impact of jobs who is not moved on the total waiting time, and $\Delta_5 = \max(0, w_5^2) - \max(0, w_5)$ to compute the effect of the moved job on the total waiting time.

Now, let's start the proven process. Beginning with the unique role of the queue leader, we list all possible situations and discuss the details of each situation.

- **Case 1:** $w_4 \le 0, w_5 > p_4 - w_4$, under those conditions, job 4 is the queue leader, and once job 4 is moved forward, the waiting time of job 5 is positive, implying job 5 is one of the members of the continuous queue.
- **Case 2:** $w_4 \le 0, 0 < w_5 \le p_4 - w_4$, in this case, job 4 is the queue leader and job 5 is part of the continuous queue at the beginning. After job 4 moves forward, job 5 becomes the new queue leader.
- **Case 3:** $w_4 \le 0, w_5 \le 0$, both jobs 4 and 5 are the queue leader, after job 4 moves forward, job 5 still be the queue leader.
- **Case 4:** $w_4 > 0, w_5 > p_4$, here, both jobs 4 and 5 belong to the continuous queue, and once job 4 moves, job 5's state is unchanged.

- **Case 5:** $w_4 > 0, 0 < w_5 \leq p_4$, at this point, jobs 4 and 5 are part of the continuous queue. Job 5 is the new queue leader after job 4's relocation.
- **Case 6:** $w_4 > 0, w_5 \leq 0$, now, job 4 belongs to the continuous queue and job 5 is the queue leader. After job 4 moves forward, job 5 still be the queue leader.

We have listed the six cases above. Now, we will use mathematical formulas to deduce the impact of job 4's movement on the queue and its mathematical laws.

Case 1: $w_4 \leq 0, w_5 > p_4 - w_4$

Under those conditions, job 4 is the queue leader, and once job 4 is moved forward, the waiting time of job 5 is positive, implying job 5 is one of the members of the continuous queue. So we have:

$$\begin{aligned} w_5^2 &= w_5 + \Delta w_5 \\ &= w_5 - (p_4 - w_4) > 0, \end{aligned}$$

so job 5 remains in the continuous queue and the impact on the total waiting time is

$$\Delta_5 = -\Delta w_6^- = -(p_4 - w_4).$$

The new waiting time of job 4 is

$$\begin{aligned} w_4^2 &= r_5 + w_5^2 + p_5 - r_4 \\ &= r_5 + w_5 - p_4 + w_4 + p_5 - r_4 \\ &= w_4 + p_5. \end{aligned}$$

(1) If $w_4 + p_5 \leq 0$, meaning job 4 still be the queue leader after its relocation: then

$$\begin{aligned} \Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= 0. \end{aligned}$$

The waiting time change for job 6 is

$$\begin{aligned} \Delta w_6 &= r_4^o + p_4 - r_6 - w_6 \\ &= r_4^o + p_4 - r_6 - (r_5 + w_5 + p_5 - r_6) \\ &= r_4^o - r_5 + p_4 - w_5 - p_5 \\ &= -p_5. \end{aligned}$$

Thus, the total waiting time changes as follows:

$$\begin{aligned} \Delta w &= \Delta_4 + \Delta_5 + \pi_5(-p_5) \\ &= p_5 - (p_4 - w_4) - p_5 + \pi_6(-p_5). \end{aligned}$$

(2) If $w_4 + p_5 > 0$, job 4 remains in the continuous queue:

The waiting time of job 4 is increased by $w_4 + p_5$, so $\Delta_4 = w_4 + p_5$. The change of job 6's waiting time is

$$\begin{aligned} \Delta w_6 &= r_4 + w_4^2 + p_4 - r_6 - w_6 \\ &= r_4 + w_4 + p_5 + p_4 - r_6 - (r_5 + w_5 + p_5 - r_6) \\ &= r_4 + w_4 + p_4 - w_5 - r_5 \\ &= w_4. \end{aligned}$$

Finally, the total waiting time change is

$$\begin{aligned} \Delta w &= \Delta_4 + \Delta_5 + \pi_5(w_4) \\ &= w_4 + p_5 - (p_4 - w_4) + \pi_6(w_4) \end{aligned}$$

$$= w_4 + p_5 - p_4 + w_4 + \pi_6(w_4).$$

Case 2: $w_4 \le 0, 0 < w_5 \le p_4 - w_4$

In this case, job 4 is the queue leader and job 5 is part of the continuous queue at the beginning. After job 4 moves forward, job 5 becomes the new queue leader.

At this situation,

$$\begin{aligned} w_5^2 &= w_5 - (p_4 - w_4) \le 0, \\ \Delta_5 &= -\Delta w_6^- = -w_5, \end{aligned}$$

job 5 becomes the new queue leader and decreases the total waiting time by $\Delta_5 = -w_5$. The updated waiting time of job 4 is:

$$\begin{aligned} w_4^2 &= r_5^o + p_5 - r_4 \\ &= r_5^o + p_5 - r_4 - p_4 + p_4 \\ &= p_4 + p_5 - w_5. \end{aligned}$$

(1) If $p_4 + p_5 - w_5 \le 0$, it means job 4 still be the queue leader after the movement of it:

For the reason that $w_4 \le 0$ and $w_4^2 \le 0$, so job 4 does not affect the total waiting time $\Delta_4 = 0$. The adjustment of job 6's waiting time is

$$\begin{aligned} \Delta w_6 &= r_4^o + p_4 - r_6 - w_6 \\ &= r_4^o + p_4 - r_6 - (r_5 + w_5 + p_5 - r_6) \\ &= w_5 - w_5 - p_5 \\ &= -p_5. \end{aligned}$$

Finally, the total waiting time change is

$$\begin{aligned} \Delta w &= \Delta_4 + \Delta_5 + \pi_5(-p_5) \\ &= p_5 - w_5 - p_5 + \pi_6(-p_5). \end{aligned}$$

(2) If $p_4 + p_5 - w_5 > 0$, job 4 is one of the members of the continuous queue:

From the condition $w_4 \le 0$ and $w_4^2 = p_4 + p_5 - w_5$, we can get

$$\begin{aligned} \Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= p_4 + p_5 - w_5. \end{aligned}$$

The waiting time adjustment for job 6 is

$$\begin{aligned} \Delta w_6 &= r_4 + w_4^2 + p_4 - r_6 - w_6 \\ &= r_4 + p_4 + p_5 - w_5 + p_4 - r_6 - (r_5 + w_5 + p_5 - r_6) \\ &= r_4 + p_4 - 2w_5 + p_4 - r_5 \\ &= p_4 - w_5. \end{aligned}$$

Consequently, the total waiting time changes as follows:

$$\begin{aligned} \Delta w &= \Delta_4 + \Delta_5 + \pi_6(p_4 - w_5) \\ &= p_4 + p_5 - w_5 - w_5 + \pi_6(p_4 - w_5) \\ &= p_5 - w_5 + p_4 - w_5 + \pi_6(p_4 - w_5). \end{aligned}$$

Case 3: $w_4 \le 0, w_5 \le 0$

Both jobs 4 and 5 are the queue leader, after job 4 moves forward, job 5 still be the queue leader.

At this stage, we can get the new waiting time of job 5 as follows:

$$w_5^2 = w_5 - (p_4 - w_4) \le 0,$$

$$\Delta_5 = -\Delta w_6^- = 0,$$

both the waiting time of job 5 is non-positive, job 5 still be the queue leader, and the impact on the total waiting time is $\Delta_5 = 0$. The updated waiting time of job 4 is:

$$\begin{aligned} w_4^2 &= r_5^o + p_5 - r_4 \\ &= p_4 + p_5 - w_5. \end{aligned}$$

Since $w_5 \le 0$, it follows that $w_4^2 > 0$. Besides, $w_4 \le 0$, so we can get that

$$\begin{aligned} \Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= p_4 + p_5 - w_5. \end{aligned}$$

The new waiting time of job 6 is calculated as follows:

$$\begin{aligned} \Delta w_6 &= r_4 + w_4^2 + p_4 - r_6 - w_6 \\ &= r_4 + p_4 + p_5 - w_5 + p_4 - r_6 - (r_5 + p_5 - r_6) \\ &= r_4 + p_4 - w_5 + p_4 - r_5 \\ &= p_4. \end{aligned}$$

In the end, the total waiting time change is:

$$\begin{aligned} \Delta w &= \Delta_4 + \Delta_5 + \pi_6(p_4) \\ &= p_5 - w_5 + p_4 + \pi_6(p_4). \end{aligned}$$

Case 4: $w_4 > 0, w_5 > p_4$

Here, both jobs 4 and 5 belong to the continuous queue, and once job 4 moves, job 5's state is unchanged.

Under this condition,

$$\begin{aligned} w_5^2 &= w_5 - \Delta w_5 \\ &= w_5 - p_4 > 0, \end{aligned}$$

job 5 is the members of the continuous queue, and the impact on the total waiting time is

$$\Delta_5 = -\Delta w_6^- = -p_4.$$

The updated waiting time of job 4 is

$$\begin{aligned} w_4^2 &= r_5 + w_5^2 + p_5 - r_4 \\ &= r_5 - p_4 + w_5 + p_5 - r_4 \\ &= p_5 + w_4. \end{aligned}$$

Because $w_4 > 0$, it is obvious that $w_4^2 > 0$, job 4 is one of the members of the continuous queue, and its contribution to the total waiting time is

$$\begin{aligned} \Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= p_5 \end{aligned}$$

For job 6, the waiting time change is

$$\begin{aligned} \Delta w_6 &= r_4 + w_4^2 + p_4 - r_6 - w_6 \\ &= r_4 + p_5 + w_4 + p_4 - r_6 - (r_5 + w_5 + p_5 - r_6) \\ &= r_4 + w_4 + p_4 - r_5 - w_5 \\ &= 0. \end{aligned}$$

Consequently, the total waiting time change is

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \pi_6(0) \\ &= p_5 - p_4 + \pi_6(0).\end{aligned}$$

Case 5: $w_4 > 0, 0 < w_5 \leq p_4$

At this point, jobs 4 and 5 are part of the continuous queue. Job 5 is the new queue leader after job 4's relocation.

In this case,

$$\begin{aligned}w_5^2 &= w_5 - \Delta w_5 \\ &= w_5 - p_4 \leq 0, \\ \Delta_5 &= -\Delta w_6^- = -w_5.\end{aligned}$$

Since job 5 becomes the new queue leader, the impact on the total waiting time is $\Delta_5 = -w_5$. The new waiting time of job 4, after its relocation is

$$\begin{aligned}w_4^2 &= r_5^o + p_5 - r_4 \\ &= r_5^o + p_5 - r_4 - p_4 + p_4 + w_4 - w_4 \\ &= p_4 + p_5 - w_5 + w_4.\end{aligned}$$

Because $w_5 \leq p_4, w_4 > 0$, it has that $w_4^2 > 0$ and job 4 in the continuous queue. The contribution of job 4 to the total waiting time is as follows:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= p_4 + p_5 - w_5.\end{aligned}$$

The adjustment of job 6 can be calculated as follows:

$$\begin{aligned}\Delta w_6 &= r_5^o + p_5 + p_4 - r_6 - w_6 \\ &= r_5^o + p_5 + p_4 - r_6 - (r_5^o + w_5 + p_5 - r_6) \\ &= p_4 - w_5.\end{aligned}$$

Finally, the total waiting time change is

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \pi_6(p_4 - w_5) \\ &= p_5 - w_5 + p_4 - w_5 + \pi_6(p_4 - w_5).\end{aligned}$$

Case 6: $w_4 > 0, w_5 \leq 0$

Now, job 4 belongs to the continuous queue and job 5 is the queue leader. After job 4 moves forward, job 5 still be the queue leader.

In this case,

$$\begin{aligned}w_5^2 &= w_5 - \Delta w_5 \\ &= w_5 - p_4 \leq 0, \\ \Delta_5 &= -\Delta w_6^- = 0,\end{aligned}$$

job 5 still be the queue leader, and has no effect on the total waiting time. The new waiting time of job 4 is calculated as follows:

$$\begin{aligned}w_4^2 &= r_5^o + p_5 - r_4 \\ &= r_5^o + p_5 - r_4 + p_4 - p_4 + w_4 - w_4 \\ &= p_4 + p_5 + w_4 - w_5.\end{aligned}$$

Because $w_5 \leq 0, w_4 > 0$, $w_4^2 > 0$. The impact on the total waiting time is

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= w_4^2 - w_4 \\ &= p_4 + p_5 - w_5.\end{aligned}$$

For job 6, the waiting time adjustment is

$$\begin{aligned}\Delta w_6 &= r_5^o + p_5 + p_4 - r_6 - w_6 \\ &= r_5^o + p_5 + p_4 - r_6 - (r_5^o + p_5 - r_6) \\ &= p_4.\end{aligned}$$

Finally, the change of the total waiting time is

$$\begin{aligned}\Delta w &= p_4 - w_5 + p_5 + \pi_6(p_4) \\ &= p_5 - w_5 + p_4 + \pi_6(p_4).\end{aligned}$$

So far, we have completed the discussion of all cases. In the next step, we will summarize the underlying laws. First, we gather the outcomes and observe their patterns.

Case 1: $w_4 \leq 0, w_5 > p_4 - w_4$

$$w_4^2 = w_4 + p_5.$$

(1) If $w_4 + p_5 \leq 0$, then

$$\Delta w = p_5 - (p_4 - w_4) - p_5 + \pi_6(-p_5).$$

(2) If $w_4 + p_5 > 0$, then

$$\Delta w = w_4 + p_5 - p_4 + w_4 + \pi_6(w_4).$$

Case 2: $w_4 \leq 0, 0 < w_5 \leq p_4 - w_4$

$$w_4^2 = p_4 + p_5 - w_5.$$

(1) If $p_4 + p_5 - w_5 \leq 0$, then:

$$\Delta w = p_5 - w_5 - p_5 + \pi_6(-p_5).$$

(2) If $p_4 + p_5 - w_5 > 0$, then:

$$\Delta w = p_5 - w_5 + p_4 - w_5 + \pi_6(p_4 - w_5).$$

Case 3: $w_4 \leq 0, w_5 \leq 0$

$$\begin{aligned}w_4^2 &= p_4 + p_5 - w_5, \\ \Delta w &= p_5 - w_5 + p_4 + \pi_6(p_4).\end{aligned}$$

Case 4:$w_4 > 0, w_5 > p_4$

$$\begin{aligned}w_4^2 &= p_5 + w_4, \\ \Delta w &= p_5 - p_4 + \pi_6(0).\end{aligned}$$

Case 5: $w_4 > 0, 0 < w_5 \leq p_4$

$$\begin{aligned}w_4^2 &= p_4 + p_5 - w_5 + w_4, \\ \Delta w &= p_5 - w_5 + p_4 - w_5 + \pi_6(p_4 - w_5).\end{aligned}$$

Case 6: $w_4 > 0, w_5 \leq 0$

$$\begin{aligned}w_4^2 &= p_4 + p_5 + w_4 - w_5, \\ \Delta w &= p_5 - w_5 + p_4 + \pi_6(p_4).\end{aligned}$$

After organizing the results, we can observe that the terms in $\pi(.)$ always have corresponding terms in the first half of the formula, and these terms are closely related to the conditions that segment scenarios. Besides, it is obvious that the remaining terms also connect to the splitting conditions. We call the terms in $\pi(.)$ as the latter terms I_2 and the remaining terms as the former terms I_1. Based on this observations, we summarize I_1 and I_2 separately as follows.

We start by summarizing I_2:

We combine conditions (1) and (2) for I_2 as follows:

Case 1: $w_4 \leq 0, w_5 > p_4 - w_4$

(1) If $w_4 + p_5 \leq 0$:

$$I_2 = -p_5.$$

(2) If $w_4 + p_5 > 0$:

$$I_2 = w_4.$$

Combining (1) and (2), it is obvious that we take the larger of $w_4, -p_5$, yielding:

$$I_2 = \max(w_4, -p_5).$$

Case 2:$w_4 \leq 0, 0 < w_5 \leq p_4 - w_4$

(1) If $p_4 + p_5 - w_5 \leq 0$:

$$I_2 = -p_5.$$

(2) If $p_4 + p_5 - w_5 > 0$:

$$I_2 = p_4 - w_5.$$

From the splitting conditions (1)(2), we can get that I_2 always take the larger of $p_4 - w_5, -p_5$, so:

$$I_2 = \max(p_4 - w_5, -p_5).$$

Case 3: $w_4 \leq 0, w_5 \leq 0$

$$I_2 = p_4.$$

Case 4: $w_4 > 0, w_5 > p_4$

$$I_2 = 0.$$

Case 5: $w_4 > 0, 0 < w_5 \leq p_4$

$$I_2 = p_4 - w_5.$$

Case 6: $w_4 > 0, w_5 \leq 0$

$$I_2 = p_4.$$

Integrate the case when $0 < w_5$:

Cases 1, 2: $w_4 \leq 0, 0 < w_5$

Both cases share the term $-p_5$, with differing terms $p_4 - w_5$ and w_4. Under the given conditions, in Case 1, we have $w_5 > p_4 - w_4$, and in Case 2, we have $0 < w_5 \leq p_4 - w_4$, it is clear that I_2 takes the largest of $p_4 - w_5$ and w_4. Thus we get:

$$I_2 = \max(p_4 - w_5, w_4, -p_5).$$

Case 3: $w_4 \leq 0, w_5 \leq 0$

$$I_2 = p_4.$$

Case 4, 5: $w_4 > 0, 0 < w_5$

Based on the conditions of Case 4 $w_5 > p_4$ and Case 5 $0 < w_5 \leq p_4$, I_2 takes the larger of $p_4 - w_5$ and 0:

$$I_2 = \max(p_4 - w_5, 0).$$

Case 6: $w_4 > 0, w_5 \leq 0$

$$I_2 = p_4.$$

Combine the conditions related to w_4.

Cases 1, 2, 4, 5: $0 < w_5$

It compares w_4 with 0, leading to:

$$I_2 = \max(p_4 - w_5, \min(0, w_4), -p_5).$$

Cases 3, 6: $w_5 \leq 0$

$$I_2 = p_4.$$

Based on the preceding analysis, the mathematical formula for I_2 can be expressed as follows:

$$I_2 = \begin{cases} \max(p_4 - w_5, \min(w_4, 0), -p_5), & if w_5 > 0, \\ p_4, & if\ w_5 \leq 0. \end{cases}$$

Similarly, the conditions for the previous term I_1 are merged as follows: First, summarizing the different cases of I_1, we have:

Case 1: $w_4 \leq 0, w_5 > p_4 - w_4$

$$I_1 = p_5 - (p_4 - w_4).$$

Case 2: $w_4 \leq 0, 0 < w_5 \leq p_4 - w_4$

$$I_1 = p_5 - w_5.$$

Case 3: $w_4 \leq 0, w_5 \leq 0$

$$I_1 = p_5 - w_5.$$

Case 4: $w_4 > 0, w_5 > p_4$

$$I_1 = p_5 - p_4.$$

Case 5: $w_4 > 0, 0 < w_5 \leq p_4$

$$I_1 = p_5 - w_5.$$

Case 6: $w_4 > 0, w_5 \leq 0$

$$I_1 = p_5 - w_5.$$

Now, let us integrate the conditions under $0 < w_5$:

Cases 1, 2: $w_4 \leq 0, w_5 > 0$

In both cases, I_1 shares the common term p_5. The differing term in Case 1 is $w_5 > p_4 - w_4$, taking $p_4 - w_4$, while in Case 2, it is $0 < w_5 \leq p_4 - w_4$,taking w_5. So, I_1 always takes the minimum of w_5 and $p_4 - w_4$:

$$I_1 = p_5 - \min(p_4 - w_4, w_5)$$

Case 3: $w_4 \leq 0, w_5 \leq 0$

$$I_1 = p_5 - w_5.$$

Cases 4, 5: $w_4 > 0, w_5 > 0$

Similarly, I_1 takes the smaller of p_4 and w_5

$$I_1 = p_5 - \min(p_4, w_5).$$

Case 6: $w_4 > 0, w_5 \leq 0$

$$I_1 = p_5 - w_5.$$

Finally, combining the conditions that $w_4 \leq 0$ and $w_4 > 0$:

Cases 1, 2, 4, 5: $w_5 > 0$

Then, we compare o with w_4, taking the smaller of the two.

$$I_1 = p_5 - \min(p_4 - \min(0, w_4), w_5).$$

Cases 3, 6: $w_5 \leq 0$

The expression for I_1 is:

$$I_1 = p_5 - w_5 = p_5 - \min(0, w_5).$$

According to Lemma 1, $\min(p_4 - \min(0, w_4), w_5)$ and 0 is actually equivalent to:

$$\Delta w_6^- = \begin{cases} 0, & \textit{if } w_5 \leq 0, \\ \min(p_4 - \min(0, w_4), w_5), & \textit{if } w_5 > 0. \end{cases}$$

And $\min(0, w_5)$ takes 0 when $w_5 \leq 0$, else takes w_5. So, I_1 can be expressed as:

$$I_1 = p_5 - \Delta w_6^- - \min(0, w_5).$$

In summary, we have obtained the formula for the change in total waiting time Δw when job 4 is relocated between jobs 5 and 6:

$$\Delta w = I_1 + I_2 + \pi_6(I_2),$$

where

$$I_1 = p_5 - \Delta w_6^- - \min(0, w_5),$$

and initial value is $\Delta w_5^- = p_4 - \min(0, w_4)$.

$$I_2 = \begin{cases} p_4, & \textit{if } w_5 \leq 0, \\ \max\left(\min(w_4, 0), -p_5, p_4 - w_5\right), & \textit{if } w_5 > 0. \end{cases}$$

However, we can compute the updated waiting times for all jobs in the queue except for the moved job 4, when we know the initial conditions $\Delta w_5^- = p_4 - \min(0, w_4)$ and I_2, and using the recursive formulations of Lemma 1 and Lemma 2. To complete the full information of the queue, it is necessary to summarize the updated waiting time w_4^2 of job 4.

To begin, we list the calculated outcomes for w_4^2:

Case 1: $w_4 \leq 0, w_5 > p_4 - w_4$

$$w_4^2 = w_4 + p_5.$$

Case 2: $w_4 \leq 0, 0 < w_5 \leq p_4 - w_4$

$$w_4^2 = p_4 + p_5 - w_5.$$

Case 3: $w_4 \leq 0, w_5 \leq 0$

$$w_4^2 = p_4 + p_5 - w_5.$$

Case 4: $w_4 > 0, w_5 > p_4$

$$w_4^2 = p_5 + w_4.$$

Case 5: $w_4 > 0, 0 < w_5 \leq p_4$

$$w_4^2 = p_4 + p_5 - w_5 + w_4.$$

Case 6: $w_4 > 0, w_5 \leq 0$

$$w_4^2 = p_4 + p_5 - w_5 + w_4.$$

Analogous to the condition combination process for I_1, it follows:

Cases 1, 2, 3: $w_4 \leq 0$

$$w_4^2 = p_5 + p_4 - \max(0, \min(p_4 - w_4, w_5)) - \min(0, w_5).$$

Cases 4, 5, 6: $w_4 > 0$

$$w_4^2 = p_5 + p_4 - \max(0, \min(p_4, w_5)) - \min(0, w_5) + w_4.$$

Finally, integrating the conditions for w_4, we have:

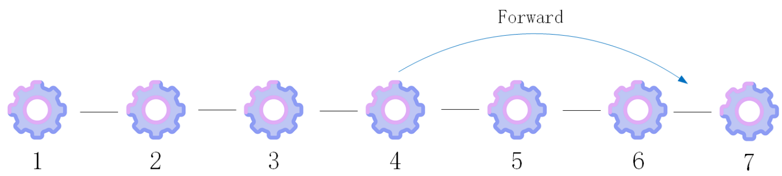

Fig. 2 Fig 2. Job 4 moves to the position between jobs 6 and 7.

Cases 1, 2, 3, 4, 5, 6:

$$\begin{aligned} w_4^2 &= p_5 + p_4 - \max(0, \min(p_4 - \min(0, w_4), w_5)) - \min(0, w_5) + \max(0, w_4) \\ &= p_5 + p_4 - \Delta w_6^- - \min(0, w_5) + \max(0, w_4). \end{aligned}$$

We summarize all the information about job 4 relocating to the position between jobs 5 and 6. The equation for the change in total waiting time is as follows:

$$\Delta w = I_1 + I_2 + \pi_6(I_2),$$

where

$$I_1 = p_5 - \Delta w_6^- - \min(0, w_5),$$

and initial value is $\Delta w_5^- = p_4 - \min(0, w_4)$.

$$I_2 = \begin{cases} p_4, & \textit{if } w_5 \le 0, \\ \max(\min(w_4, 0), -p_5, p_4 - w_5), & \textit{if } w_5 > 0. \end{cases}$$

The updated waiting time of job 4 is:

$$w_4^2 = p_4 + p_5 - \Delta w_6^- - \min(0, w_5) + \max(0, w_4).$$

where initial value is $\Delta w_5^- = p_4 - \min(0, w_4)$.

We have achieved a breakthrough from “0” to “1”, by obtaining the formula above. But does this formula still hold when job 4 moves to a more forward position? Or, in other words, what is the law for extending this formula? Thus, if we want to generalize it to an infinite scope, it is necessary to examine and broaden this formula from “1” to “2”. We examine whether the law holds by shifting job 4 to the location between jobs 6 and 7 in the next step, as illustrated in Figure 2.

Similar to the case of job 4 moving to the position between jobs 5 and 6, we also start from the special role of the queue leader and discuss it by cases:

- **Case 1:** $w_4 \le 0, w_5 > p_4 - w_4, w_6 > p_4 - w_4$, initially, job 4 is the queue leader, and jobs 5 and 6 are in the contiguous queue. After moving job 4, jobs 5 and 6 still be in the contiguous queue.
- **Case 2:** $w_4 \le 0, w_5 > p_4 - w_4, 0 < w_6 \le p_4 - w_4$, in this case, job 4 is the queue leader, and jobs 5 and 6 are in the contiguous queue. Once job 4 is moved, job 5 remains in the contiguous queue, while job 6 becomes the new queue leader.
- **Case 3:** $w_4 \le 0, w_5 > p_4 - w_4, w_6 \le 0$, jobs 4 and 6 are the queue leader, and job 5 is one of the members of the contiguous queue, in the beginning. After job 4 is moved, jobs 5 and 6 are not changed.
- **Case 4:** $w_4 \le 0, 0 < w_5 \le p_4 - w_4, w_6 > w_5$, originally, job 4 is the queue leader, and jobs 5 and 6 are in the continuous queue. Following job 4's movement, job 5 becomes the new queue leader, while job 6 remains in the contiguous queue.

- **Case 5:** $w_4 \leq 0, 0 < w_5 \leq p_4 - w_4, w_6 \leq 0$, at this point, jobs 4 and 6 are the queue leader, and job 5 is in the continuous queue. Once job 4 is relocated, both jobs 5 and 6 are the new queue leader.
- **Case 6:** $w_4 \leq 0, 0 < w_5 \leq p_4 - w_4, w_6 \leq 0$, in this case, jobs 4 and 6 are the queue leader, and job 5 is part of the continuous queue. Once job 4 is relocated, both jobs 5 and 6 become the new queue leader.
- **Case 7:** $w_4 \leq 0, w_5 \leq 0, w_6 > 0$, in this stage, jobs 4 and 5 are the queue leader, and job 6 is in the contiguous queue. After moving job 4, jobs 5 and 6 are unchanged.
- **Case 8:** $w_4 \leq 0, w_5 \leq 0, w_6 \leq 0$, under this condition, jobs 4,5, and 6 are the queue leader. Following job 4's movement, jobs 5 and 6 still be the queue leader.
- **Case 9:** $w_4 > 0, w_5 > p_4, w_6 > p_4$, initially, jobs 4,5, and 6 are part of the continuous queue. When job 4 is relocated, jobs 5 and 6 remain unchanged.
- **Case 10:** $w_4 > 0, w_5 > p_4, 0 < w_6 \leq p_4$, in this case, jobs 4,5, and 6 are in the continuous queue. Once job 4 is moved, job 5 is unchanged, while job 6 shifts to the new queue leader.
- **Case 11:** $w_4 > 0, w_5 > p_4, w_6 \leq 0$, here, jobs 4 and 5 are part of the continuous queue, and job 6 is the queue leader. Once job 4 is moved, jobs 5 and 6 remain their role.
- **Case 12:** $w_4 > 0, 0 < w_5 \leq p_4, w_6 > w_5$, initially, jobs 4, 5, and 6 are in the continuous queue. Once job 4's relocation, job 5 becomes the new queue leader, while job 6 is unchanged.
- **Case 13:** $w_4 > 0, 0 < w_5 \leq p_4, 0 < w_6 \leq w_5$, under this condition, jobs 4, 5, and 6 are one of the members of the continuous queue. Following job 4's movement, jobs 5 and 6 become the new queue leader.
- **Case 14:** $w_4 > 0, 0 < w_5 \leq p_4, w_6 \leq 0$, at this point, jobs 4 and 5 are in the continuous queue, while job 6 is the queue leader. After job 4's relocation, jobs 5 and 6 are the queue leader.
- **Case 15:** $w_4 > 0, w_5 \leq 0, w_6 > 0$, at this stage, jobs 4 and 6 are part of the continuous queue, and job 5 is the queue leader. After moving job 4, jobs 5 and 6 are unchanged.
- **Case 16:** $w_4 > 0, w_5 \leq 0, w_6 \leq 0$, job 4 is in the contiguous queue, while jobs 5 and 6 are queue heads, in the beginning. Following job 4's movement, jobs 5 and 6 remain the queue leader.

Case 1: $w_4 \leq 0, w_5 > p_4 - w_4, w_6 > p_4 - w_4$

Initially, job 4 is the queue leader, and jobs 5 and 6 are in the contiguous queue. After moving job 4, jobs 5 and 6 still be in the contiguous queue. In this case, we can get from the conditions that:

$$\Delta_5 = -\Delta w_6^- = -(p_4 - w_4),$$
$$\Delta_6 = -\Delta w_7^- = -(p_4 - w_4),$$

The new waiting time of job 4 is:

$$\begin{aligned} w_4^2 &= r_6 + w_6^2 + p_6 - r_4 \\ &= r_6 - r_5 + r_5 + w_4 - p_4 + w_6 + p_6 - r_4 \\ &= p_5 + w_5 - w_6 + w_4 - p_4 + w_6 + p_6 + p_4 - w_5 \\ &= p_5 + p_6 + w_4. \end{aligned}$$

(1) if $p_5 + p_6 + w_4 \leq 0$, job 4 is the queue leader, we can obtain follows:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4) = 0,\\ \Delta w_7 &= r_4^o + p_4 - r_7 - w_7\\ &= r_4^o + p_4 - r_7 - (r_4^o + p_4 + p_5 + p_6 - r_7)\\ &= -(p_5 + p_6).\end{aligned}$$

The change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= -2(p_4 - w_4) + p_5 + p_6 - (p_5 + p_6) + \pi_7(-(p_5 + p_6)).\end{aligned}$$

(2) if $p_5 + p_6 + w_4 > 0$, job 4 is in the continuous queue:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4)\\ &= p_5 + p_6 + w_4.\end{aligned}$$

$$\begin{aligned}\Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7\\ &= r_4 + p_5 + p_6 + w_4 + p_4 - r_7 - (r_4 + p_4 + p_5 + p_6 - r_7)\\ &= w_4.\end{aligned}$$

The adjustment of total waiting time is as follows:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - 2(p_4 - w_4) + w_4 + \pi_7(w_4).\end{aligned}$$

Case 2: $w_4 \leq 0, w_5 > p_4 - w_4, 0 < w_6 \leq p_4 - w_4$

In this case, job 4 is the queue leader, and jobs 5 and 6 are in the contiguous queue. Once job 4 is moved, job 5 remains in the contiguous queue, while job 6 becomes the new queue leader. Under this conditions:

$$\begin{aligned}\Delta_5 &= -\Delta w_6^- = -(p_4 - w_4),\\ \Delta_6 &= -\Delta w_7^- = -w_6,\end{aligned}$$

The updated waiting time of job 4 is:

$$\begin{aligned}w_4^2 &= r_6^o + p_6 - r_4\\ &= r_6^o - r_4 - p_4 - p_5 + p_4 + p_5 + p_6\\ &= p_4 + p_5 + p_6 - w_6.\end{aligned}$$

(1) If $p_4 + p_5 + p_6 - w_6 \leq 0$, job 4 still be the queue leader after its relocation, it follows:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = 0,$$

$$\begin{aligned}\Delta w_7 &= r_4^o + p_4 - r_7 - w_7\\ &= r_4^o + p_4 - r_7 - (r_4^o + p_4 + p_5 + p_6 - r_7)\\ &= -(p_5 + p_6).\end{aligned}$$

At this point, the impact on the total waiting time is formulated as follows:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 + (w_4 - p_4) - w_6 - (p_5 + p_6) + \pi_7(-(p_5 + p_6)).\end{aligned}$$

(2) If $p_4 + p_5 + p_6 - w_6 > 0$, job 4 is one of the members of the continuous queue, we can obtain that:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4)$$

$$= p_4 + p_5 + p_6 - w_6,$$

$$\begin{aligned}\Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7\\ &= r_4 + p_4 + p_5 + p_6 - w_6 + p_4 - r_7 - (r_4^o + p_4 + p_6 + p_6 - r_7)\\ &= p_4 - w_6.\end{aligned}$$

Now, the impact on the total waiting time is formulated as follows:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - w_6 - (p_4 - w_4) + p_4 - w_6 + \pi_7(p_4 - w_6).\end{aligned}$$

Case 3: $w_4 \le 0, w_5 > p_4 - w_4, w_6 \le 0$

Jobs 4 and 6 are the queue leader, and job 5 is one of the members of the contiguous queue, in the beginning. After job 4 is moved, jobs 5 and 6 not changed. This moment

$$\begin{aligned}\Delta_5 &= -\Delta w_6^- = -(p_4 - w_4),\\ \Delta_6 &= -\Delta w_7^- = 0.\end{aligned}$$

The new waiting time of job 4 is

$$\begin{aligned}w_4^2 &= r_6^o + p_6 - r_4\\ &= r_6^o + p_6 - r_4 - p_4 - p_5 + p_4 + p_5\\ &= p_4 + p_5 + p_6 - w_6.\end{aligned}$$

$\because w_6 \le 0, \therefore w_4^2 > 0$, so we have:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4)\\ &= p_4 + p_5 + p_6 - w_6.\end{aligned}$$

$$\begin{aligned}\Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7\\ &= r_4 + p_4 + (p_4 + p_5 + p_6 - w_6) - r_7 - (r_4 + p_4 + p_5 + p_6 - w_6 - r_7)\\ &= p_4.\end{aligned}$$

In the end, the total waiting time calculated as follows:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - (p_4 - w_4) - w_6 + p_4 + \pi_7(p_4).\end{aligned}$$

Case 4: $w_4 \le 0, 0 < w_5 \le p_4 - w_4, w_6 > w_5$

Originally, job 4 is the queue leader, and jobs 5 and 6 are in the continuous queue. Following job 4's movement, job 5 becomes the new queue leader, while job 6 remains in the contiguous queue. Now, we can get that:

$$\begin{aligned}\Delta_5 &= -\Delta w_6^- = -w_5,\\ \Delta_6 &= -\Delta w_7^- = -w_5.\end{aligned}$$

The w_4^2 is calculated:

$$\begin{aligned}w_4^2 &= r_5^o + p_5 + p_6 - r_4\\ &= p_4 + p_5 + p_6 - w_5.\end{aligned}$$

(1) When $p_4 + p_5 + p_6 - w_5 \le 0$, job 4 is the queue leader, following job's waiting time is:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4) = 0,\\ \Delta w_7 &= r_4^o + p_4 - r_7 - w_7\end{aligned}$$

$$
\begin{aligned}
&= r_4^o + p_4 - r_7 - (r_4^o + p_4 + p_5 + p_6 - r_7) \\
&= -(p_5 + p_6).
\end{aligned}
$$

Thus, the total waiting time change can be expressed:

$$
\begin{aligned}
\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7) \\
&= p_5 + p_6 - w_5 - w_5 - (p_5 + p_6) + \pi_7(-(p_5 + p_6)).
\end{aligned}
$$

(2) When $p_4 + p_5 + p_6 - w_5 > 0$, job 4 becomes part of the continuous queue.

$$
\begin{aligned}
\Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\
&= p_4 + p_5 + p_6 - w_5. \\
\Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7 \\
&= r_4 + p_4 + p_5 + p_6 - w_5 + p_4 - r_7 - w_7 \\
&= p_4 - w_5.
\end{aligned}
$$

Finally, the change of the total waiting time is:

$$
\begin{aligned}
\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7) \\
&= p_5 + p_6 - 2w_5 + (p_4 - w_5) + \pi_7(p_4 - w_5).
\end{aligned}
$$

Case 5: $w_4 \le 0, 0 < w_5 \le p_4 - w_4, 0 < w_6 \le w_5$

At this point, jobs 4 and 6 are the queue leader, and job 5 is in the continuous queue. Once job 4 is relocated, both jobs 5 and 6 are the new queue leader. In this stage, the impact of jobs 5 and 6 on the total waiting time is

$$
\begin{aligned}
\Delta_5 &= -\Delta w_6^- = -w_5, \\
\Delta_6 &= -\Delta w_7^- = -w_6.
\end{aligned}
$$

The new waiting time of job 4 is

$$
\begin{aligned}
w_4^2 &= r_6^o + p_6 - r_4 \\
&= p_6 + p_4 + p_5 - w_6.
\end{aligned}
$$

(1) If $p_6 + p_4 + p_5 - w_6 \le 0$, job 4 is the queue leader.

$$
\begin{aligned}
\Delta_4 &= \max(0, w_4^2) - \max(0, w_4) = 0 \\
\Delta w_7 &= r_4^o + p_4 - r_7 - w_7 \\
&= r_4^o + p_4 - r_7 - (r_4^o + p_4 + p_5 + p_6 - r_7) \\
&= -(p_5 + p_6).
\end{aligned}
$$

In summary, the adjustment of total waiting time is:

$$
\begin{aligned}
\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7) \\
&= p_5 + p_6 - w_5 - w_6 - (p_5 + p_6) + \pi_7(-(p_5 + p_6)).
\end{aligned}
$$

(2) If $p_6 + p_4 + p_5 - w_6 > 0$, then job 4 is within the continuous queue.

$$
\begin{aligned}
\Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\
&= p_6 + p_4 + p_5 - w_6. \\
\Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7 \\
&= r_4 + p_4 + p_5 + p_6 - w_6 + p_4 - r_7 - (r_4 + p_4 + p_5 + p_6 - r_7) \\
&= p_4 - w_6.
\end{aligned}
$$

Finally, the change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - w_5 - w_6 + (p_4 - w_6) + \pi_7(p_4 - w_6).\end{aligned}$$

Case 6: $w_4 \le 0, 0 < w_5 \le p_4 - w_4, w_6 \le 0$

In this case, jobs 4 and 6 are the queue leader, and job 5 is part of the continuous queue. Once job 4 is relocated, both jobs 5 and 6 become the new queue leader. Under this condition, it can be obtained as follows:

$$\begin{aligned}\Delta_5 &= -\Delta w_6^- = -w_5,\\ \Delta_6 &= -\Delta w_7^- = 0.\end{aligned}$$

The new waiting time of job 4, following its movement is:

$$\begin{aligned}w_4^2 &= r_6^o + p_6 - r_4\\ &= p_4 + p_5 + p_6 - w_6.\end{aligned}$$

$\because w_6 \le 0 \therefore w_4^2 > 0.$

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4)\\ &= p_4 + p_5 + p_6 - w_6.\\ \Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7\\ &= r_4 + p_4 - r_7 + p_4 + p_5 + p_6 - w_6 - w_7\\ &= p_4.\end{aligned}$$

The total waiting time change is:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - w_5 - w_6 + p_4 + \pi_7(p_4).\end{aligned}$$

Case 7: $w_4 \le 0, w_5 \le 0, w_6 > 0$

In this stage, jobs 4 and 5 are the queue leader, and job 6 is in the contiguous queue. After moving job 4, jobs 5 and 6 are unchanged. In this situation, we can get that:

$$\begin{aligned}\Delta_5 &= -\Delta w_6^- = 0,\\ \Delta_6 &= -\Delta w_7^- = 0.\end{aligned}$$

The updated waiting time is:

$$\begin{aligned}w_4^2 &= r_5^o + p_5 + p_6 - r_4\\ &= p_4 + p_5 + p_6 - w_5.\end{aligned}$$

Since $w_5 \le 0$, we have $w_4^2 > 0$.

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4)\\ &= p_4 + p_5 + p_6 - w_5.\\ \Delta w_7 &= r_4^o + p_4 + p_5 + p_6 - r_7 - w_7\\ &= p_4.\end{aligned}$$

Thus, the adjustment of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - w_5 + p_4 + \pi_7(p_4).\end{aligned}$$

Case 8: $w_4 \le 0, w_5 \le 0, w_6 \le 0$

Under this condition, jobs 4,5, and 6 are the queue leader. Following job 4's movement, jobs 5 and 6 still be the queue leader. Now,

$$\Delta_5 = -\Delta w_6^- = 0,$$
$$\Delta_6 = -\Delta w_7^- = 0.$$

The updated waiting time is:

$$\begin{aligned} w_4^2 &= r_6^o + p_6 - r_4 \\ &= r_6^o + p_6 - r_5 + r_5 - r_4 \\ &= p_4 + p_5 + p_6 - w_5 - w_6. \end{aligned}$$

$\because w_5 \leq 0, w_6 \leq 0, \therefore w_4^2 > 0.$

$$\begin{aligned} \Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= p_4 + p_5 + p_6 - w_5 - w_6. \\ \Delta w_7 &= r_4^o + p_4 + p_5 + p_6 - r_7 - w_7 \\ &= p_4. \end{aligned}$$

Finally, the total waiting time change is:

$$\begin{aligned} \Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7) \\ &= p_5 + p_6 - w_5 - w_6 + p_4 + \pi_7(p_4). \end{aligned}$$

Case 9: $w_4 > 0, w_5 > p_4, w_6 > p_4$

Initially, jobs 4,5, and 6 are part of the continuous queue. When job 4 is relocated, jobs 5 and 6 remain unchanged. In this case, we have:

$$\Delta_5 = -\Delta w_6^- = -p_4,$$
$$\Delta_6 = -\Delta w_7^- = -p_4.$$

The updated waiting time is:

$$\begin{aligned} w_4^2 &= r_6 + w_6^2 + p_6 - r_4 \\ &= r_6 + w_6 - p_4 + p_6 - r_4 - w_4 + w_4 - p_5 + p_5 \\ &= p_5 + p_6 + w_4. \end{aligned}$$

$\because w_4 > 0, \therefore w_4^2 > 0.$

$$\begin{aligned} \Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= p_5 + p_6. \\ \Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7 \\ &= r_4 + p_5 + p_6 + w_4 + p_4 - r_7 - w_7 \\ &= 0. \end{aligned}$$

The change of total waiting time is:

$$\begin{aligned} \Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7) \\ &= p_5 + p_6 - 2p_4 + \pi_7(0). \end{aligned}$$

Case 10: $w_4 > 0, w_5 > p_4, 0 < w_6 \leq p_4$

In this case, jobs 4,5, and 6 are in the continuous queue. Once job 4 is moved, job 5 is unchanged, while job 6 shifts to the new queue leader. At this point, it follows:

$$\Delta_5 = -\Delta w_6^- = -p_4,$$

$$\Delta_6 = -\Delta w_7^- = -w_6.$$

The updated waiting time is:

$$\begin{aligned} w_4^2 &= r_6^o + p_6 - r_4 \\ &= r_6^o + p_6 - r_4 - w_4 + w_4 - p_4 + p_4 - p_5 + p_5 \\ &= p_4 + p_5 + p_6 + w_4 - w_6. \end{aligned}$$

Since $w_6 \leq p_4, w_4 > 0$, we have $w_4^2 > 0$, so:

$$\begin{aligned} \Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= w_4^2 - w_4 = p_4 + p_5 + p_6 - w_6. \end{aligned}$$

$$\begin{aligned} \Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7 \\ &= r_4 + p_4 + p_5 + p_6 - w_6 + w_4 + p_4 - r_7 - w_7 \\ &= p_4 - w_6. \end{aligned}$$

Finally, the total waiting time change is:

$$\begin{aligned} \Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7) \\ &= p_5 + p_6 - (p_4 + w_6) + (p_4 - w_6) + \pi_7(p_4 - w_6). \end{aligned}$$

Case 11: $w_4 > 0, w_5 > p_4, w_6 \leq 0$

Here, jobs 4 and 5 are part of the continuous queue, and job 6 is the queue leader. Once job 4 is moved, jobs 5 and 6 remain their role. In this case, we get:

$$\Delta_5 = -\Delta w_6^- = -p_4,$$
$$\Delta_6 = -\Delta w_7^- = 0.$$

The new waiting time of job 4 is calculated as follows:

$$\begin{aligned} w_4^2 &= r_6^o + p_6 - r_4 \\ &= r_6^o + p_6 - r_4 - w_4 + w_4 - p_4 + p_4 - p_5 + p_5 \\ &= p_4 + p_5 + p_6 + w_4 - w_6. \end{aligned}$$

Under conditions $w_4 > 0, w_6 \leq 0$, we can get $w_4^2 > 0$, so:

$$\begin{aligned} \Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= p_4 + p_5 + p_6 - w_6. \end{aligned}$$

$$\begin{aligned} \Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7 \\ &= r_4 + p_4 + p_5 + p_6 + w_4 + p_4 - r_7 - w_7 \\ &= p_4. \end{aligned}$$

In the end, we have the formula of the total waiting time change is as follows:

$$\begin{aligned} \Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7) \\ &= p_5 + p_6 - (p_4 + w_6) + p_4 + \pi_7(p_4). \end{aligned}$$

Case 12: $w_4 > 0, 0 < w_5 \leq p_4, w_6 > w_5$

Initially, jobs 4, 5, and 6 are in the continuous queue. Once job 4's relocation, job 5 becomes the new queue leader, while job 6 is unchanged. This moment, we have:

$$\Delta_5 = -\Delta w_6^- = -w_5,$$
$$\Delta_6 = -\Delta w_7^- = -w_5.$$

The new waiting of job 4 is:

$$\begin{aligned}w_4^2 &= r_5^o + p_5 + p_6 - r_4\\ &= p_4 + p_5 + p_6 + w_4 - w_5.\end{aligned}$$

$\because p_4 > w_5, \therefore w_4^2 > 0$, so:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4)\\ &= p_4 + p_5 + p_6 - w_5.\\ \Delta w_7 &= r_4 + w_4^2 + p_4 - r_7 - w_7\\ &= r_4 + p_4 + p_5 + p_6 + w_4 - w_5 + p_4 - r_7 - (r_4 + w_4 + p_4 + p_5 + p_6 - r_7)\\ &= p_4 - w_5.\end{aligned}$$

In summary, the change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - 2w_5 + (p_4 - w_5) + \pi_7(p_4 - w_5).\end{aligned}$$

Case 13: $w_4 > 0, 0 < w_5 \leq p_4, 0 < w_6 \leq w_5$

Under this condition, jobs 4, 5, and 6 are one of the members of the continuous queue. Following job 4's movement, jobs 5 and 6 become the new queue leader. It follows:

$$\Delta_5 = -\Delta w_6^- = -w_5,$$
$$\Delta_6 = -\Delta w_7^- = -w_6.$$

The new waiting time of job 4 is:

$$\begin{aligned}w_4^2 &= r_6^o + p_6 - r_4\\ &= p_4 + p_5 + p_6 + w_4 - w_6.\end{aligned}$$

$\because p_4 \geq w_5 \geq w_6, w_4 > 0 \therefore w_4^2 > 0$. So:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4)\\ &= p_4 + p_5 + p_6 - w_6.\\ \Delta w_7 &= r_6^o + p_4 + p_6 - w_7\\ &= r_6^o + p_4 + p_6 - (r_6 + w_6 + p_6 - r_7)\\ &= p_4 - w_6.\end{aligned}$$

Consequently, the change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - w_5 - w_6 + (p_4 - w_6) + \pi_7(p_4 - w_6).\end{aligned}$$

Case 14: $w_4 > 0, 0 < w_5 \leq p_4, w_6 \leq 0$

At this point, jobs 4 and 5 are in the continuous queue, while job 6 is the queue leader. After job 4's relocation, jobs 5 and 6 are the queue leader. In this situation, it follows:

$$\Delta_5 = -\Delta w_6^- = -w_5,$$
$$\Delta_6 = -\Delta w_7^- = 0.$$

The new waiting time of job 4 is:

$$\begin{aligned}w_4^2 &= r_6^o + p_6 - r_4\\ &= r_6^o + p_6 - r_4 - w_4 + w_4 - p_5 + p_5 - p_4 + p_4\end{aligned}$$

$$= p_4 + p_5 + p_6 + w_4 - w_6.$$

When we have those conditions: $w_6 \le 0, w_4 > 0$, it follows that $w_4^2 > 0$:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4)\\ &= p_4 + p_5 + p_6 - w_6.\\ \Delta w_7 &= r_6^o + p_6 + p_4 - w_7\\ &= r_6^o + p_6 + p_4 - (r_6^o + p_6 - r_7)\\ &= p_4.\end{aligned}$$

Finally, we have:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - w_5 - w_6 + p_4 + \pi_7(p_4).\end{aligned}$$

Case 15: $w_4 > 0, w_5 \le 0, w_6 > 0$

At this stage, jobs 4 and 6 are part of the continuous queue, and job 5 is the queue leader. After moving job 4, jobs 5 and 6 are unchanged. It can be deducted that:

$$\begin{aligned}\Delta_5 &= -\Delta w_6^- = 0,\\ \Delta_6 &= -\Delta w_7^- = 0.\end{aligned}$$

The new waiting time of job 4 is:

$$\begin{aligned}w_4^2 &= r_5^o + p_5 + p_6 - r_4\\ &= r_5^o + p_5 + p_6 - r_4 - p_4 + p_4 - w_4 + w_4\\ &= p_4 + p_5 + p_6 + w_4 - w_5.\end{aligned}$$

Since $w_4 > 0, w_5 \le 0$, we have $w_4^2 > 0$, it follows that:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4)\\ &= p_4 + p_5 + p_6 - w_5.\\ \Delta w_7 &= r_5^o + p_5 + p_6 + p_4 - r_7 - w_7\\ &= p_4.\end{aligned}$$

Thus, the total waiting time change is calculated as follows:

$$\begin{aligned}\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7)\\ &= p_5 + p_6 - w_5 + p_4 + \pi_7(p_4).\end{aligned}$$

Case 16: $w_4 > 0, w_5 \le 0, w_6 \le 0$

Job 4 is in the contiguous queue, while jobs 5 and 6 are queue heads, in the beginning. Following job 4's movement, jobs 5 and 6 remain the queue leader. In this case, it follows that:

$$\begin{aligned}\Delta_5 &= -\Delta w_6^- = 0,\\ \Delta_6 &= -\Delta w_7^- = 0.\end{aligned}$$

The updated time of job 4 is:

$$\begin{aligned}w_4^2 &= r_6^o + p_6 - r_4\\ &= r_6 + p_6 - r_4 - w_4 - p_4 - p_5 + w_5 + p_4 + p_5 - w_5\\ &= p_4 + p_5 + p_6 + w_4 - w_6 - w_5.\end{aligned}$$

$\because w_4 > 0, w_5 \le 0, w_6 \le 0$, we have $w_4^2 > 0$, it follows that:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4)$$

$$
\begin{aligned}
&= p_4 + p_5 + p_6 - w_5 - w_6. \\
\Delta w_7 &= r_6 + p_6 + p_4 - r_7 - w_7 \\
&= p_4.
\end{aligned}
$$

Finally, we have the change of total waiting time is:

$$
\begin{aligned}
\Delta w &= \Delta_4 + \Delta_5 + \Delta_6 + \pi_7(\Delta w_7) \\
&= p_5 + p_6 - w_5 - w_6 + p_4 + \pi_7(p_4).
\end{aligned}
$$

In the same way, we consolidate and integrate w_4, I_2, and I_1 following the approach used for the job 4 forward movement only one position.

First, let us integrate the conditions for I_2:

Cases 1, 2, 4, 5, 9, 10, 12, and 13 $w_5 > 0, w_6 > 0$

$$I_2 = \max(\min(0, w_4), -(p_5 + p_6), p_4 - w_5, p_4 - w_6).$$

Case 3: $w_4 \leq 0, w_5 > p_4 - w_4, w_6 \leq 0$

$$I_2 = p_4.$$

Case 6: $w_4 \leq 0, 0 < w_5 \leq p_4 - w_4, w_6 \leq 0$

$$I_2 = p_4.$$

Case 7: $w_4 \leq 0, w_5 \leq 0, w_6 > 0$

$$I_2 = p_4.$$

Case 8: $w_4 \leq 0, w_5 \leq 0, w_6 \leq 0$

$$I_2 = p_4.$$

Case 11: $w_4 > 0, w_5 > p_4, w_6 \leq 0$

$$I_2 = p_4.$$

Case 14: $w_4 > 0, 0 < w_5 \leq p_4, w_6 \leq 0$

$$I_2 = p_4.$$

Case 15: $w_4 > 0, w_5 \leq 0, w_6 > 0$

$$I_2 = p_4.$$

Case 16: $w_4 > 0, w_5 \leq 0, w_6 \leq 0$

$$I_2 = p_4.$$

From observation, it is obvious that when either $w_5 \leq 0$ or $w_6 \leq 0$, we have that:

$$I_2 = p_4.$$

If both them are greater than 0:

$$
\begin{aligned}
I_2 &= \max(\min(w_4, 0), -(p_5 + p_6), p_4 - w_5, p_4 - w_6) \\
&= \max\left(\min(w_4, 0), -(\sum_{k=5}^{6} p_k), \max_{k \in 5,6}(p_4 - w_k)\right).
\end{aligned}
$$

Therefore, I_2 can be expressed mathematically as:

$$
I_2 = \begin{cases} p_4, & \textit{otherwise}, \\ \max\left(\min(w_4, 0), -(\sum_{k=5}^{6} p_k), \max_{k \in 5,6}(p_4 - w_k)\right), & \textit{if } \boldsymbol{w_k} \succ 0, k \in \{5, 6\}. \end{cases}
$$

Likewise, we proceed to combine the conditions for the I_1:

Cases 1, 2, 4, 5, 9, 10, 12, 13: $w_5 > 0, w_6 > 0$

$$I_1 = p_5 + p_6 - \min(p_4 - \min(w_4, 0), w_5) - \min(p_4 - \min(w_4, 0), w_5, w_6).$$

Cases 3, 6, 11, 14: $w_5 > 0, w_6 \leq 0$

$$I_1 = p_5 + p_6 - \min(p_4 - \min(w_4, 0), w_5) - \min(w_6, 0).$$

Cases 7, 15: $w_5 \leq 0, w_6 > 0$

$$I_1 = p_5 + p_6 - \min(w_5, 0).$$

Cases 8, 16: $w_5 \leq 0, w_6 \leq 0$

$$I_1 = p_5 + p_6 - \min(w_5, 0) - \min(w_6, 0).$$

If we omit $\min(w_5, 0)$ and $\min(w_6, 0)$, it can observe:

- When $w_5 > 0$ and $w_6 > 0$, we have

$$\begin{aligned} \Delta w_6^- &= \min(p_4 - \min(w_4, 0), w_5), \\ \Delta w_7^- &= \min(p_4 - \min(w_4, 0), w_5, w_6). \end{aligned}$$

- When $w_5 > 0$ and $w_6 \leq 0$, leading to:

$$\begin{aligned} \Delta w_6^- &= \min(p_4 - \min(w_4, 0), w_5), \\ \Delta w_7^- &= 0. \end{aligned}$$

- When $w_5 \leq 0$, the term is $\Delta w_6^- = 0$ and $\Delta w_7^- = 0$.

Obviously, it is conforms to the law of Lemma 1. So, I_1 can be expressed as:

$$\begin{aligned} I_1 &= p_5 + p_6 - (\Delta w_6^- + \min(w_5, 0)) - (\Delta w_7^- + \min(w_6, 0)) \\ &= \sum_{k=5}^{6} (p_k - \min(w_k, 0)) - \sum_{k=6}^{7} \Delta w_k^-. \end{aligned}$$

In the next step, we combine the conditions for w_4^2:

Cases 1, 2, 4, 5, 9, 10, 12, 13:$w_5 > 0, w_6 > 0$

$$w_4^2 = p_4 + p_5 + p_6 - \min(w_5, w_6, p_4 - \min(0, w_4)) + \max(0, w_4).$$

Cases 3, 6, 7, 8, 11, 14, 15, 16: $w_6 \leq 0$

$$w_4^2 = p_4 + p_5 + p_6 - \min(0, w_5) - \min(0, w_6) + \max(0, w_4).$$

According to the regularity of Lemma 1, it is obviously that $\Delta w_7^- = \min(w_5, w_6, p_4 - \min(0, w_4))$ when $w_5 > 0$ and $w_6 > 0$. When either w_5 or w_6 is less than or equal to 0, $\Delta w_7^- = 0$. So, we have:

$$w_4^2 = p_4 + p_5 + p_6 - \Delta w_7^- - \min(0, w_5) - \min(0, w_6) + \max(0, w_4).$$

Conclusion: Now, that we have obtained all the information for job 4 moving to a position between jobs 6 and 7, the calculation formula for the change of total waiting time is:

$$\Delta w = I_1 + I_2 + \pi_7(I_2).$$

where

$$I_1 = \sum_{k=5}^{6} (p_k - \min(0, w_k)) - \sum_{k=6}^{7} \Delta w_k^-,$$

$$I_2 = \begin{cases} p_4, & \textit{otherwise}, \\ \max\left(\min(w_4, 0), -(\sum_{k=5}^{6} p_k), \max_{k\in\{5,6\}}(p_4 - w_k)\right), & \textit{if } \boldsymbol{w_k} \succ 0, k \in \{5, 6\}. \end{cases}$$

The new waiting time of job 4 is:

$$w_4^2 = \sum_{k=4}^{6} p_k - \sum_{k=5}^{6} \min(0, w_5) - \Delta w_7^- + \max(0, w_4).$$

And the initial value is $\Delta w_5^- = p_4 - \min(0, w_4)$.

So far, we have established the law of the job moves forward. It can be described as: Given the current job sequence σ_t, if job $\sigma_t(i)$ is moved forward to the position immediately after job $\sigma_t(k)$ (where $k \in [i+1, n]$), the change in total waiting time Δw follows this formula:

$$\Delta w = I_1 + I_2 + \pi_{\sigma_t(k+1)}(I_2),$$

where

$$I_1 = \sum_{j=i+1}^{k} \left(p_{\sigma_t(j)} - \min(0, w_{\sigma_t(j)})\right) - \sum_{j=i+2}^{k+1} \Delta w^-_{\sigma_t(j)},$$

$$I_2 = \begin{cases} p_{\sigma(i)}, & \text{otherwise}, \\ \max\Big(\min(0, w_{\sigma(i)}), \quad -\sum_{j=i+1}^{k} p_{\sigma(j)}, \ \max_{j\in[i+1,k]} (p_{\sigma(i)} - w_{\sigma(j)})\Big), & \text{if } \mathbf{w}_{\sigma(j)} \succ \mathbf{0}, \forall j \in [i+1, k]. \end{cases}$$

Furthermore, after the relocation of job $\sigma_t(i)$, its new waiting time $w^2_{\sigma_t(i)}$ is calculated as:

$$w^2_{\sigma_t(i)} = \sum_{j=i}^{k} p_{\sigma_t(j)} - \sum_{j=i+1}^{k} \min(0, w_{\sigma_t(j)}) - \Delta w^-_{\sigma_t(k+1)} + \max(0, w_{\sigma_t(i)}).$$

And $\Delta w^-_{\sigma_t(j)}$ follows the recursive formula given in Lemma 1, with the initial value:

$$\Delta w^-_{\sigma_t(i+1)} = p_{\sigma_t(i)} - \min(0, w_{\sigma_t(i)}).$$

□

A.5 Proof of Theorem 2

Proof The proof of Theorem 2 is analogous to that of Theorem 1. Consider an arbitrary schedule σ_t and any four consecutive jobs $\sigma_t(i)$, $\sigma_t(i+1)$, $\sigma_t(i+2)$, and $\sigma_t(i+3)$, with $i \in 1, \ldots, n-3$. We start from the simplest case, in which job $\sigma_t(i+2)$ is moved to the position immediately preceding $\sigma_t(i+1)$, as shown in Figure 1. Without loss of generality, denote these four jobs by 2, 3, 4, and 5, respectively; that is, let $\sigma_t(i) = 2$, $\sigma_t(i+1) = 3$, $\sigma_t(i+2) = 4$, and $\sigma_t(i+3) = 5$. This relabeling is adopted purely for notational simplicity and has no effect on the validity of the subsequent analysis.

Inspired by the proof of Theorem 1, we adjust the formula toward the form of Δw_j^+. The notation is the same as the proof of Theorem 1. In the same way, $\Delta_5 = \Delta w_6^+$ is used to calculate the impact of non-moving jobs on the total waiting time, while $\Delta_5 = \max(0, w_5^2) - \max(0, w_5)$ calculates the effect of the moving job. Beginning with the special role of the queue leader, we classify the scenarios.

Case 1: $w_4 \leq 0, w_3 \leq 0$

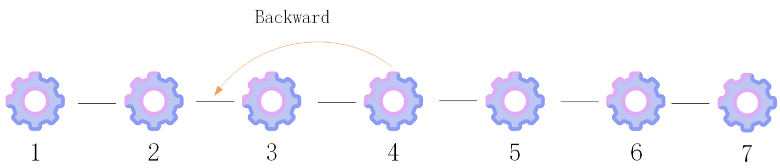

Fig. 3 Fig 3. Job 4 is moved to the position between jobs 3 and 4.

Initially, both jobs 3 and 4 are the queue leader. In this case, we can calculate the change of job 4:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\ &= r_0^o + p_0 - r_4 - (r_0^o + p_0 + p_3 - r_4) \\ &= w_3 - p_3.\end{aligned}$$

The updated waiting time of job 4 is:

$$\begin{aligned}w_4^2 &= w_4 + \Delta w_4 \\ &= w_3 + w_4 - p_3.\end{aligned}$$

Since $w_4 \le 0, w_3 < 0$, we can get $w_4^2 \le 0$. Then, job 4 still be the queue leader after its relocation. The impact of job 4 on the total waiting time is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = 0.$$

The updated information of job 3 is as follows:

$$\begin{aligned}\Delta w_3 &= r_4^o + p_4 - r_3 - w_3 \\ &= p_3 + p_4 - w_4. \\ w_3^2 &= w_3 + \Delta w_3 \\ &= p_3 + p_4 + w_3 - w_4 > 0. \\ \Delta_3 &= \Delta w_4^+ = p_3 + p_4 - w_4.\end{aligned}$$

Because $w_3^2 > 0$, job 3 is part of the continuous queue. The change of job 5 is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= r_3 + p_3 + p_4 + w_3 - w_4 + p_3 - r_5 - (r_3 + w_3 + p_3 + p_4 - r_5 - w_4) \\ &= p_3.\end{aligned}$$

The change of total waiting time is calculated as below:

$$\begin{aligned}\Delta w &= \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 - w_4 + p_3 + \pi_5(p_3).\end{aligned}$$

Case 2: $w_4 \le 0, w_3 > 0$

Job 4 is the queue leader, while job 3 is part of the continuous queue. Here, the change waiting time for job 4, and its new waiting time are, respectively:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\ &= r_0^o + p_0 - r_4 - (r_0^o + p_0 + p_3 - r_4) \\ &= -p_3. \\ w_4^2 &= w_4 + \Delta w_4\end{aligned}$$

$$= w_4 - p_3 \leq 0.$$

For $w_4^2 \leq 0$, job 4 is the queue leader even its movement. The impact of job 4 on the total waiting time is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = 0.$$

The new information about job 3 is:

$$\begin{aligned}\Delta w_3 &= r_4^o + p_4 - r_3 - w_3 \\ &= p_3 - w_4 + p_4. \\ w_3^2 &= w_3 + \Delta w_3 \\ &= p_3 + p_4 + w_3 - w_4 > 0. \\ \Delta_3 &= \Delta w_4^+ = p_3 + p_4 - w_4.\end{aligned}$$

For $w_3^2 > 0$, job 3 is in the continuous queue. The change of job 5's waiting time is as follows:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= r_3 + p_3 + w_3 - w_4 + p_3 - r_5 + p_4 - (r_3 + w_3 + p_3 + p_4 - r_5 - w_4) \\ &= p_3.\end{aligned}$$

The impact on the total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 - w_4 + p_3 + \pi_5(p_3).\end{aligned}$$

Case 3: $w_4 > 0, w_3 \leq 0$

Job 4 is one of the members of the continuous queue, job 3 is the queue leader. Under this condition, the change of job 4's waiting time, and its new waiting time are, respectively:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\ &= r_0^o + p_0 - r_4 - (r_0^o + p_0 - w_3 + p_3 - r_4) \\ &= w_3 - p_3. \\ w_4^2 &= w_4 + \Delta w_4 \\ &= w_3 + w_4 - p_3.\end{aligned}$$

Comparing 0 with w_4^2:

(1) If $w_3 + w_4 - p_3 \leq 0$, job 4 is the queue leader, and its impact on the total waiting time is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = -w_4.$$

The information about job 3, after job 4's relocation, is listed:

$$\begin{aligned}\Delta w_3 &= r_4^o + p_4 - r_3 - w_3 \\ &= r_4^o + p_4 - r_3 - w_3 + p_3 - p_3 \\ &= p_3 + p_4 - w_3 - w_4. \\ w_3^2 &= \Delta w_3 + w_3 \\ &= p_3 + p_4 - w_4.\end{aligned}$$

It is necessary to compare 0 with w_3^2:

1) When $p_3 + p_4 - w_4 \leq 0$, job 3 is the queue leader. The impact of job 3 on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = 0.$$

The adjustment of job 5 is as follows:

$$\begin{aligned}\Delta w_5 &= r_3^o + p_3 - r_5 - w_5 \\ &= r_3^o + p_3 - r_5 - (r_3^o + p_3 + p_4 - r_5) \\ &= -p_4.\end{aligned}$$

The change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 - w_4 - p_4 + \pi_5(-p_4).\end{aligned}$$

2) When $p_3 + p_4 - w_4 > 0$, it means job 3 is part of the continuous queue. The impact of job 3' waiting time change on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = p_3 + p_4 - w_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= r_3^o + p_3 + p_4 - w_4 + p_3 - r_5 - (r_3^o + p_3 + p_4 - r_5) \\ &= p_3 - w_4.\end{aligned}$$

The change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 - w_4 + p_3 - w_4 + \pi_5(p_3 - w_4).\end{aligned}$$

(2) If $w_3 + w_4 - p_3 > 0$, job 4 is one of the members of the continuous queue, its impact on the total waiting time is:

$$\Delta_4 = w_3 - p_3.$$

The new information of job 3 is listed:

$$\begin{aligned}\Delta w_3 &= r_4^o + w_4^2 + p_4 - r_3 - w_3 \\ &= r_4 + w_4 + w_3 - p_3 + p_4 - r_3 - w_3 \\ &= r_3^o + p_3 - w_4 + w_4 + w_3 - p_3 + p_4 - r_3 - w_3 \\ &= p_4. \\ w_3^2 &= \Delta w_3 + w_3 \\ &= w_3 + p_4.\end{aligned}$$

It needs to compare 0 with w_3^2 as well.

1) When $w_3 + p_4 > 0$, job 3 is in the continuous queue, its impact on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = w_3 + p_4.$$

The change of job 5 is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= r_3 + w_3 + p_4 + p_3 - r_5 - (r_3^o + p_3 + p_4 - r_5) \\ &= w_3.\end{aligned}$$

Then, the adjustment of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 - p_3 + w_3 + w_3 + \pi_5(w_3).\end{aligned}$$

2) When $w_3 + p_4 \leq 0$, job 3 is the queue leader, it changes the total waiting time by:

$$\Delta_3 = \Delta w_4^+ = 0.$$

The change of job 5 is:

$$\begin{aligned}\Delta w_5 &= r_3^o + p_3 - r_5 - w_5 \\ &= r_3^o + p_3 - r_5 - (r_3 + p_3 + p_4 - r_5) \\ &= -p_4.\end{aligned}$$

The change of total waiting time is as follows:

$$\begin{aligned}\Delta w &= \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= w_3 - p_3 + p_4 - p_4 + \pi_5(-p_4).\end{aligned}$$

Case 4: $w_4 > 0, w_3 > 0$, both jobs 3 and 4 are the members of the continuous queue. At this point, the change of job 4's waiting time is:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\ &= r_0^o + p_0 - r_4 - (r_0^O + p_0 + p_3 - r_4) \\ &= -p_3.\end{aligned}$$

The updated waiting time of job 4 is:

$$\begin{aligned}w_4^2 &= w_4 + \Delta w_4 \\ &= w_4 - p_3.\end{aligned}$$

It needs to know whether job 4 is the queue leader, therefore comparing 0 with w_4^2.

(1) If $w_4 - p_3 > 0$, job 4 is part of the continuous queue, while the change of total waiting time caused by job 4 is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = -p_3.$$

The information about job 3, once job 4 is moved, is:

$$\begin{aligned}\Delta w_3 &= r_4 + w_4^2 + p_4 - r_3 - w_3 \\ &= r_4 + w_4 - p_3 + p_4 - r_3 - w_3 \\ &= p_4. \\ w_3^2 &= \Delta w_3 + w_3 \\ &= w_3 + p_4 > 0.\end{aligned}$$

For $w_3^2 > 0$, job 3 is in the continuous queue, and its impact on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = p_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= r_3 + w_3 + p_4 + p_3 - r_5 - (r_3 + w_3 + p_3 + p_4 - r_5) \\ &= 0.\end{aligned}$$

The total waiting time is changed as follows:

$$\begin{aligned}\Delta w &= \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 - p_3 + \pi_5(0).\end{aligned}$$

(2) If $w_4 - p_3 \leq 0$, job 4 becomes the queue leader, and the total waiting change caused by job 4 is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = -w_4.$$

The new information of job 3 is as follows:

$$\begin{aligned}\Delta w_3 &= r_4^o + p_4 - r_3 - w_3 \\ &= r_3 + w_3 + p_3 - w_4 + p_4 - r_3 - w_3 \\ &= p_3 + p_4 - w_4. \\ w_3^2 &= w_3 + \Delta w_3 \\ &= p_3 + p_4 - w_4 + w_3 > 0.\end{aligned}$$

Because $w_3^2 > 0$, job 3 is one of the members of the continuous queue, and its impact on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = p_3 + p_4 - w_4.$$

Then, the change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 - w_4 + p_3 - w_4 + \pi_5(p_3 - w_4).\end{aligned}$$

Following the way of Theorem 1's proof, we partition the formula according to the items in $\pi(.)$, denoting the terms in $\pi(.)$ as I_4 and the remaining terms as I_3.

To begin, we summarize I_4 to get the expression:

$$I_4 = \max(-p_4, \min(w_3, 0), p_3 - \max(w_4, 0)).$$

I_3 doesn't fully depend on the splitting condition and needs to move toward the form of Δw_j^+. Here, we provide the detailed process.

Case 1: $w_4 \leq 0, w_3 \leq 0$

$$I_3 = p_4 - w_4.$$

Case 2: $w_4 \leq 0, w_3 > 0$

$$I_3 = p_4 - w_4.$$

Case 3: $w_4 > 0, w_3 \leq 0$

(1) If $w_3 + w_4 - p_3 \leq 0$:

$$I_3 = p_4 - w_4.$$

(2) If $w_3 + w_4 - p_3 > 0$:

$$I_3 = p_4 - p_3 + w_3.$$

Case 4: $w_4 > 0, w_3 > 0$

(1) If $w_4 - p_3 > 0$:

$$I_3 = p_4 - p_3.$$

(2) If $w_4 - p_3 \leq 0$:

$$I_3 = p_4 - w_4.$$

Combining the conditions for (1) and (2):

Case 1: $w_4 \leq 0, w_3 \leq 0$

$$I_3 = p_4 - w_4.$$

Case 2: $w_4 \leq 0, w_3 > 0$

$$I_3 = p_4 - w_4.$$

Case 3: $w_4 > 0, w_3 \leq 0$

In this case, it compares $-w_4$ with $w_3 - p_3$, and I_3 takes the greater of them:

$$I_3 = p_4 + \max(-p_3 + w_3, -w_4).$$

Case 4: $w_4 > 0, w_3 > 0$

In the same way, I_3 gets the bigger one of $-p_3$ and $-w_4$:

$$I_3 = p_4 + \max(-p_3, -w_4).$$

As mentioned at the beginning, the formula's form should be similar to Δw_j^+. We perform an equivalent reformulation of I_3 for each case:

Case 1: $w_4 \leq 0, w_3 \leq 0$

$$\begin{aligned} I_3 &= p_4 - w_4 \\ &= p_4 - w_4 + w_3 - p_3 - w_3 + p_3 \\ &= \max(p_4, p_3 + p_4 - w_3 - w_4) + w_3 - p_3. \end{aligned}$$

For $w_4 \leq 0, w_3 \leq 0$, so $p_3 + p_4 - w_3 - w_4 \geq p_4$. Thus, $\max(p_4, p_3 + p_4 - w_3 - w_4)$ always taking $p_3 + p_4 - w_3 - w_4$, above equation holds.

Case 2: $w_4 \leq 0, w_3 > 0$

$$\begin{aligned} I_3 &= p_4 - w_4 \\ &= p_4 - w_4 - p_3 + p_3 \\ &= \max(p_4, p_4 + p_3 - w_4) - p_3. \end{aligned}$$

Similarly, $w_4 \leq 0$, we can get $p_4 + p_3 - w_4 > p_4$. So $\max(p_4, p_4 + p_3 - w_4)$ always is $p_4 + p_3 - w_4$, the equation holds.

Case 3: $w_4 > 0, w_3 \leq 0$

$$\begin{aligned} I_3 &= p_4 + \max(-w_4, w_3 - p_3) \\ &= \max(p_4, p_3 + p_4 - w_3 - w_4) + w_3 - p_3. \end{aligned}$$

It only involves adjusting the terms in max by a constant, without changing their relative magnitudes, thus it will not change the result of the max operation as well.

Case 4: $w_4 > 0, w_3 > 0$

$$\begin{aligned} I_3 &= p_4 + \max(-p_3, -w_4) \\ &= \max(p_4, p_3 + p_4 - w_4) - p_3. \end{aligned}$$

Similarly to the Case 3, the equation holds.

Performing combination of similar terms leads to:

Case 1, 3: $w_3 \leq 0$

$$I_3 = \max(p_4, p_3 + p_4 - w_3 - w_4) + w_3 - p_3.$$

Case 2, 4: $w_3 > 0$

$$I_3 = \max(p_4, p_4 + p_3 - w_4) - p_3.$$

Integrating the conditions for w_3, we have:

$$\begin{aligned} I_3 &= \max(p_4, p_3 + p_4 - \min(0, w_3) - w_4) + \min(0, w_3) - p_3 \\ &= \Delta w_3^+ + \min(0, w_3) - p_3. \end{aligned}$$

Now, we have the information of I_3 and I_4, and only w_4^2 is missing to know everything. w_4^2 is summarized as:

$$w_4^2 = \min(0, w_3) + w_4 - p_3.$$

In the end, we get the formula of the change of total waiting time, when job 4 moves to the position between job 2 and 3:

$$\Delta w = I_3 + I_4 + \pi_5(I_4).$$

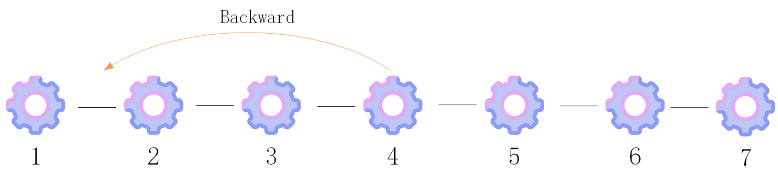

Fig. 4 Fig 4. Job 4 moves to the position between job 1 and 2.

where

$$I_3 = \Delta w_3^+ + \min(0, w_3) - p_3,$$
$$I_4 = \max(-p_4, \min(w_3, 0), p_3 - \max(w_4, 0)).$$

The initial value is $\Delta w_3^+ = \max(p_4, p_3 + p_4 - \min(0, w_3) - w_4)$.

The updated waiting time of job 4 can be calculated as:

$$w_4^2 = \min(0, w_3) - p_3 + w_4.$$

We have achieved the breakthrough from "0" to "1", we next verify the law from "1" to "2" by relocating job 4 to the position between jobs 1 and 2. There are eight cases according to the queue leader.

Case 1: $w_4 \leq 0, w_2 \leq 0, w_3 \leq 0$

When job 4 is not relocated, jobs 2, 3, and 4 are each the queue leader. In this case, the waiting time's change of job 4 and its new waiting time are respectively:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\ &= r_0^o + p_0 - r_4 - (r_0^o + p_0 - w_2 + p_2 - w_3 + p_3 - r_4) \\ &= w_2 + w_3 - p_2 - p_3. \\ w_4^2 &= w_4 + \Delta w_4 \\ &= w_2 + w_3 + w_4 - p_2 - p_3 > 0.\end{aligned}$$

For $w_4^2 > 0$, job 4 is part of the continuous queue. Once job 4 is moved, the impact of job 4 on the total waiting time is:

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\ &= w_2 + w_3 + w_4 - p_2 - p_3.\end{aligned}$$

The updated information about job 2 is:

$$\begin{aligned}\Delta w_2 &= r_4^o + p_4 - r_2 - w_2 \\ &= r_2^o + p_2 - w_3 + p_3 - w_4 + p_4 - r_2 - w_2 \\ &= p_2 + p_3 + p_4 - w_2 - w_3 - w_4. \\ w_2^2 &= w_2 + \Delta w_2 \\ &= p_2 + p_3 + p_4 - w_3 - w_4 > 0.\end{aligned}$$

In this point, job 2 is one of the members of the continuous queue. The affect of job 2 on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = p_2 + p_3 + p_4 - w_3 - w_4.$$

The new information of job 3 is:

$$\Delta w_3 = r_4^o + p_4 + p_2 - r_3 - w_3$$

$$
\begin{aligned}
&= r_4^o + p_4 + p_2 - (r_3 + p_3 - p_3) - w_3 \\
&= p_2 + p_3 + p_4 - w_4 - w_3. \\
w_3^2 &= w_3 + \Delta w_3 \\
&= p_2 + p_3 + p_4 - w_4 > 0.
\end{aligned}
$$

Because $w_3^2 > 0$, job 3 is in the continuous queue, the impact of job 3 on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = p_2 + p_3 + p_4 - w_4.$$

The change of job 5's waiting time is:

$$
\begin{aligned}
\Delta w_5 &= r_4^o + p_4 + p_3 + p_2 - r_5 - w_5 \\
&= r_4^o + p_2 + p_3 + p_4 - r_5 - (r_4^o + p_4 - r_5) \\
&= p_2 + p_3.
\end{aligned}
$$

The adjustment of total waiting time is:

$$
\begin{aligned}
\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\
&= p_2 + p_3 + p_4 - w_3 - w_4 + p_2 + p_3 + p_4 - w_4 + 0 + \pi_5(p_2 + p_3) \\
&= p_2 + p_3 + p_4 - w_3 - w_4 + p_4 - w_4 + p_2 + p_3 + \pi_5(p_2 + p_3).
\end{aligned}
$$

Case 2: $w_4 \leq 0, w_2 \leq 0, w_3 > 0$

Before job 4's relocation, both jobs 2 and 4 are the queue leader, and job 3 in the continuous queue.

The change of information about job 4 is:

$$
\begin{aligned}
\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\
&= r_0^o + p_0 - r_4 - (r_0^o + p_0 - w_2 + p_2 + p_3 - r_4) \\
&= w_2 - p_2 - p_3, \\
w_4^2 &= w_4 + \Delta w_4 \\
&= w_4 + w_2 - p_2 - p_3 < 0.
\end{aligned}
$$

Job 4 is the queue leader now, the impact of job 4 on the total waiting time is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = 0.$$

The updated waiting time of job 2 and its new waiting time are:

$$
\begin{aligned}
\Delta w_2 &= r_4^o = p_4 - r_2 - w_2. \\
w_2^2 &= w_2 + \Delta w_2 \\
&= p_2 + p_3 + p_4 - w_4 > 0.
\end{aligned}
$$

For $w_2^2 > 0$, job 2 is in the continuous queue, its impact on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = p_2 + p_3 + p_4 - w_4.$$

The new information about job 3 is:

$$
\begin{aligned}
\Delta w_3 &= r_2 + w_2^2 + p_2 - r_3 - w_3 \\
&= r_2 + p_2 + p_3 + p_4 - w_4 + p_2 - r_3 - w_3 \\
&= p_2 + p_3 + p_4 - w_4. \\
w_3^2 &= w_3 + \Delta w_3 \\
&= p_2 + p_3 + p_4 - w_4 + w_3.
\end{aligned}
$$

For $w_4 \leq 0, w_3 > 0$, so $w_3^2 > 0$ and job 3 is part of the continuous queue. The impact of job 3 on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = p_2 + p_3 + p_4 - w_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= p_2 + p_3 + p_4 - w_4 + w_4 + w_5 - p_4 - w_5 \\ &= p_2 + p_3.\end{aligned}$$

The adjustment of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_2 + p_3 + p_4 - w_4 + p_2 + p_3 + p_4 - w_4 + \pi_5(p_2 + p_3) \\ &= p_2 + p_3 + p_4 - w_4 + p_4 - w_4 + p_2 + p_3 + \pi_5(p_2 + p_3).\end{aligned}$$

Case 3: $w_4 \leq 0, w_2 > 0, w_3 \leq 0$

Before job 4 is moved, jobs 3 and 4 are the queue leader, and job 2 is part of the continuous queue. In this moment, the new waiting time of job 4 and its change of waiting time are respectively:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\ &= w_3 - p_2 - p_3, \\ w_4^2 &= w_4 + \Delta w_4 \\ &= w_3 + w_4 - p_2 - p_3 < 0.\end{aligned}$$

So, job 4 is the queue leader, and the change in total waiting time caused by job 4 is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = 0.$$

The updated information of job 2 is:

$$\begin{aligned}\Delta w_2 &= r_4^o + p_4 - r_3 - w_2 \\ &= r_4^o + p_4 - (r_2 + w_2 + p_2 - p_2 - w_3 + w_3 + p_3 - p_3) \\ &= p_4 + p_3 + p_2 - w_3 - w_4. \\ w_2^2 &= w_2 + \Delta w_2 \\ &= p_2 + p_3 + p_4 + w_2 - w_3 - w_4 > 0.\end{aligned}$$

For job 2 is in the continuous queue, its influence on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = p_2 + p_3 + p_4 - w_3 - w_4.$$

The new information for job 3 is as follows:

$$\begin{aligned}\Delta w_3 &= r_4^o + p_2 + p_4 - r_3 - w_3 \\ &= r_4^o + p_2 + p_4 - r_3 - w_3 + p_3 - p_3 \\ &= p_2 + p_3 + p_4 - w_4 - w_3. \\ w_3^2 &= w_3 + \Delta w_3 \\ &= p_2 + p_3 + p_4 - w_4 > 0.\end{aligned}$$

Job 3 is in the continuous queue for $w_3^2 > 0$, the impact on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = p_2 + p_3 + p_4 - w_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_4^o + p_4 + p_3 + p_2 - r_5 - w_5 \\ &= p_2 + p_3.\end{aligned}$$

Finally, the change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_2 + p_3 + p_4 - w_3 - w_4 + p_2 + p_3 + p_4 - w_4 + \pi_5(p_2 + p_3) \\ &= p_2 + p_3 + p_4 - w_3 - w_4 + p_4 - w_4 + p_2 + p_3 + \pi_5(p_2 + p_3).\end{aligned}$$

Case 4: $w_4 \leq 0, w_2 > 0, w_3 > 0$
Now, job 4 is the queue leader, and both job 2 and 3 are in the continuous queue. The updated information of job 4 is given by:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\ &= r_0^o + p_0 - r_4 - (r_0^o + p_0 + p_2 + p_3 - r_4) \\ &= -p_2 + p_3. \\ w_4^2 &= w_4 + \Delta w_4 \\ &= w_4 - p_2 - p_3 < 0.\end{aligned}$$

Job 4 is the queue leader, its influence on the total waiting time is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = 0.$$

The new information about job 2 is:

$$\begin{aligned}\Delta w_2 &= r_4^o + p_4 - r_2 - w_2 \\ &= r_4^o + p_4 - (w_4 - w_2 - p_2 - p_3 + r_4) - w_2 \\ &= p_2 + p_3 + p_4 - w_4. \\ w_2^2 &= w_2 + \Delta w_2 \\ &= p_2 + p_3 + p_4 - w_4 + w_2 > 0.\end{aligned}$$

For job 2 is in the continuous queue, the change in total waiting time due to job 2 is:

$$\Delta_2 = \Delta w_3^+ = p_2 + p_3 + p_4 - w_4.$$

The updated information of job 3 is:

$$\begin{aligned}\Delta w_3 &= r_4^o + p_4 + p_2 - r_3 - w_3 \\ &= r_4^o + p_4 + p_2 - r_3 - w_3 - p_3 + p_3 \\ &= p_2 + p_3 + p_4 - w_4. \\ w_3^2 &= w_3 + \Delta w_3 \\ &= p_2 + p_3 + p_4 - w_4 + w_3 > 0.\end{aligned}$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_4^o + p_2 + p_3 + p_4 - r_5 - w_5 \\ &= p_2 + p_3 + w_5 - w_5 \\ &= p_2 + p_3.\end{aligned}$$

The total waiting time's change is:

$$\Delta w = \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5)$$

$$
\begin{aligned}
&= p_2 + p_3 + p_4 - w_4 + p_2 + p_3 + p_4 - w_4 + \pi_5(p_2 + p_3) \\
&= p_2 + p_3 + p_4 - w_4 + p_4 - w_4 + p_2 + p_3 + \pi_5(p_2 + p_3).
\end{aligned}
$$

Case 5: $w_4 > 0, w_2 \leq 0, w_3 \leq 0$

Before job 4's relocation, both jobs 2 and 3 are the queue leader, and job 4 is in the continuous queue. The updated information about job 4 is:

$$
\begin{aligned}
\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\
&= r_0^o + p_0 - r_4 - (r_0^o + p_0 - r_4 - w_2 + p_2 - w_3 + p_3) \\
&= w_2 + w_3 - p_2 - p_3. \\
w_4^2 &= w_4 + \Delta w_4 \\
&= w_4 + w_3 + w_2 - p_2 - p_3.
\end{aligned}
$$

(1) If $w_4 + w_3 + w_2 - p_2 - p_3 > 0$, job 4 is part of continuous queue.

$$
\begin{aligned}
\Delta_4 &= \max(0, w_4^2) - \max(0, w_4) \\
&= w_3 + w_2 - p_2 - p_3.
\end{aligned}
$$

The new information of job 2 is:

$$
\begin{aligned}
\Delta w_2 &= r_4 + w_4^2 + p_4 - r_2 - w_2 \\
&= r_4 + w_2 + w_3 + w_4 - p_2 - p_3 + p_4 - r_2 - w_2 \\
&= p_4 + w_4 - w_4 \\
&= p_4. \\
w_2^2 &= w_2 + \Delta w_2 \\
&= w_2 + p_4.
\end{aligned}
$$

1) When $w_2 + p_4 \leq 0$, job 2 is the queue leader, the influence of job 2 on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = 0.$$

The new information of job 3 is:

$$
\begin{aligned}
\Delta w_3 &= r_2^o + p_2 - r_3 - w_3 \\
&= w_3 - w_3 = 0. \\
w_3^2 &= w_3 + \Delta w_3 \\
&= w_3 \leq 0.
\end{aligned}
$$

Then, job 3 is the queue leader, the effect of job 3 on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = 0.$$

The change of job 5's waiting time is:

$$
\begin{aligned}
\Delta w_5 &= r_3^o + p_3 - r_5 - w_5 \\
&= r_3^o + p_3 + p_4 - p_4 - r_5 - w_5 \\
&= -p_4.
\end{aligned}
$$

The adjustment of total waiting time is:

$$
\begin{aligned}
\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\
&= w_3 + w_2 - p_2 - p_3 + \pi_5(-p_4) \\
&= p_4 + w_3 + w_2 - p_2 - p_3 - p_4 + \pi_5(-p_4).
\end{aligned}
$$

2) When $w_2 + p_4 > 0$, job 2 belongs to the continuous queue, its influence on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = w_2 + p_4.$$

The updated information about job 3 is:

$$\begin{aligned}\Delta w_3 &= r_2 + w_2^2 + p_2 - r_3 - w_3\\ &= w_2^2\\ &= w_2 + p_4.\\ w_3^2 &= w_3 + \Delta w_3\\ &= w_2 + w_3 + p_4.\end{aligned}$$

i) When $w_2 + w_3 + p_4 \leq 0$, job 3 is the queue leader, and its affect on the total waiting time can be listed:

$$\Delta_3 = \Delta w_4^+ = 0.$$

The change of job 5 is:

$$\begin{aligned}\Delta w_5 &= r_3^o + p_3 - r_5 - w_5\\ &= r_3^o + p_3 + p_4 - p_4 - r_5 - w_5\\ &= -p_4.\end{aligned}$$

The change of the total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5)\\ &= p_4 + p_4 + w_2 + w_2 + w_3 - p_2 - p_3 - p_4 + \pi_5(-p_4).\end{aligned}$$

ii) When $w_2 + w_3 + p_4 > 0$, job 3 is one of the members of the continuous queue, and the change in total waiting time caused by job 3 is:

$$\Delta_3 = \Delta w_4^+ = w_2 + w_3 + p_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5\\ &= w_3^2 - p_4\\ &= w_2 + w_3.\end{aligned}$$

Finally, the change of total waiting time is as follows:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5)\\ &= p_4 + p_4 + w_2 - p_2 + w_3 - p_3 + w_2 + w_2 + w_3 + \pi_5(w_2 + w_3).\end{aligned}$$

(2) If $w_4 + w_3 + w_2 - p_2 - p_3 \leq 0$, job 4 is the queue leader, the impact of job 4 on the total waiting time is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = -w_4.$$

The updated information about job 2 is:

$$\begin{aligned}\Delta w_2 &= r_4^o + p_4 - r_2 - w_2\\ &= r_4^o + p_4 - r_2 - p_2 + p_2 - w_3 + w_3 - p_3 + p_3 - w_2\\ &= p_2 + p_3 + p_4 - w_2 - w_3 - w_4.\\ w_2^2 &= w_2 + \Delta w_2\\ &= p_2 + p_3 + p_4 - w_3 - w_4.\end{aligned}$$

1) When $p_2+p_3+p_4-w_3-w_4>0$, job 2 is part of continuous queue, its influence on the total waiting time is:

$$\Delta_2=\Delta w_3^+=p_2+p_3+p_4-w_3-w_4.$$

The information of job 3, after job 4's relocation, is as follows:

$$\begin{aligned}\Delta w_3&=r_2+w_2^2+p_2-r_3-w_3\\&=p_2+p_3+p_4-w_3-w_4.\\w_3^2&=w_3+\Delta w_3\\&=p_2+p_3+p_4-w_4.\end{aligned}$$

i) When $p_2+p_3+p_4-w_4\le 0$, job 3 is the queue leader, and its impact on the total waiting time is:

$$\Delta_3=\Delta w_4^+=0.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5&=r_3^o+p_3-r_5-w_5\\&=r_3^o+p_3+p_4-p_4-r_5-w_5\\&=-p_4.\end{aligned}$$

In the end, the change of total waiting time is:

$$\begin{aligned}\Delta w&=\Delta_2+\Delta_3+\Delta_4+\pi_5(\Delta w_5)\\&=p_2+p_3+p_4-w_3-w_4-w_4+\pi_5(-p_4)\\&=p_3+p_2+p_4-w_3-w_4+p_4-w_4-p_4+\pi_5(-p_4).\end{aligned}$$

ii) When $p_2+p_3+p_4-w_4>0$, job 3 belongs to the continuous queue, its impact on the total waiting time is:

$$\Delta_3=\Delta w_4^+=p_2+p_3+p_4-w_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5&=r_3+w_3^2+p_3-r_5-w_5\\&=r_3+w_3^2+p_3+p_4-p_4-r_5-w_5\\&=w_3^2-p_4\\&=p_2+p_3-w_4.\end{aligned}$$

The change of total waiting time is:

$$\begin{aligned}\Delta w&=\Delta_2+\Delta_3+\Delta_4+\pi_5(\Delta w_5)\\&=p_2+p_3+p_4-w_3-w_4+p_4-w_4+p_2+p_3-w_4+\pi_5(p_2+p_3-w_4).\end{aligned}$$

2) When $p_2+p_3+p_4-w_3-w_4\le 0$, job 2 is the queue leader, and its impact on the total waiting time is:

$$\Delta_2=\Delta w_3^+=0.$$

The new information about job 3, after job 4's moved, is as follows:

$$\begin{aligned}\Delta w_3&=r_2^o+p_2-r_3-w_3=0,\\w_3^2&=w_3+\Delta w_3=w_3.\end{aligned}$$

For $w_3\le 0$, job 3 still be the queue leader, and its impact on the total waiting time is:

$$\Delta_3=\Delta w_4^+=0.$$

The change of total waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3^o + p_3 - r_5 - w_5\\ &= r_3^o + p_3 + p_4 - p_4 - r_5 - w_5\\ &= -p_4.\end{aligned}$$

The adjustment of total waiting time is as follows:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5)\\ &= -w_4 + \pi_5(-p_4)\\ &= p_4 - w_4 - p_4 + \pi_5(-p_4).\end{aligned}$$

Case 6: $w_4 > 0, w_2 \leq 0, w_3 > 0$

Before job 4's movement, job 2 is the queue leader, both job 3 and 4 are in the continuous queue.

The change of job 4's waiting time and its new waiting time are, respectively:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4\\ &= r_0^o + p_0 - r_4 - (r_0^o + p_0 - w_2 + p_2 + p_3 - r_4)\\ &= w_2 - p_2 - p_3.\\ w_4^2 &= w_4 + \Delta w_4\\ &= w_2 + w_4 - p_2 - p_3.\end{aligned}$$

(1) If $w_2 + w_4 - p_2 - p_3 > 0$, job 4 is part of the continuous queue.

$$\begin{aligned}\Delta_4 &= \max(0, w_4^2) - \max(0, w_4)\\ &= w_2 - p_2 - p_3.\end{aligned}$$

The new information of job 2 is:

$$\begin{aligned}\Delta w_2 &= r_4 + w_4^2 + p_4 - r_2 - w_2\\ &= w_2 + w_4 + p_4 - p_2 - p_3 + r_4 - r_2 - p_2 + p_2 - p_3 + p_3 - w_2\\ &= w_4 + p_4 - p_2 - p_3 + p_2 + p_3 - w_4\\ &= p_4.\\ w_2^2 &= w_2 + \Delta w_2\\ &= w_2 + p_4.\end{aligned}$$

1) When $w_2 + p_4 > 0$, job 2 belongs to the continuous queue, and its influence on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = w_2 + p_4.$$

The updated information of job 3 is:

$$\begin{aligned}\Delta w_3 &= r_2 + w_2^2 + p_2 - r_3 - w_3\\ &= r_2 + w_2 + p_4 + p_2 - r_3 - w_3\\ &= w_2 + p_4.\\ w_3^2 &= w_3 + \Delta w_3\\ &= w_3 + w_2 + p_4 > 0.\end{aligned}$$

For $w_3^2 > 0$, job 3 is one of the members of the continuous queue, and its impact on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = w_2 + p_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5\\ &= r_3 + w_2 + p_4 + w_3 + p_3 - r_5 - (r_3 + w_3 + p_3 + p_4 - r_5)\\ &= w_2.\end{aligned}$$

At last, the total waiting time's change is:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5)\\ &= w_2 - p_2 - p_3 + w_2 + p_4 + p_4 + w_2 + \pi_5(w_2).\end{aligned}$$

2) When $w_2 + p_4 \leq 0$, job 2 is the queue leader, and its affect on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = 0.$$

The new information about job 3 is:

$$\begin{aligned}\Delta w_3 &= r_2^o + p_2 - r_3 - w_3\\ &= 0.\\ w_3^2 &= w_3 + \Delta w_3\\ &= w_3.\end{aligned}$$

The total waiting time's change is:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5)\\ &= w_2 - p_2 - p_3 + p_4 - p_4 + \pi_5(-p_4).\end{aligned}$$

(2) If $w_2 + w_4 - p_2 - p_3 \leq 0$, job 4 is the queue leader, we can calculate that:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = 0.$$

The updated information of job 2 is:

$$\begin{aligned}\Delta w_2 &= r_4^o + p_4 - r_2 - w_2\\ &= r_4^o + p_4 - (w_4 - p_2 - p_3 + r_4) - w_2\\ &= p_2 + p_3 + p_4 - w_4 - w_2.\\ w_2^2 &= w_2 + \Delta w_2\\ &= p_2 + p_3 + p_4 - w_4.\end{aligned}$$

1) When $p_2 + p_3 + p_4 - w_4 > 0$, job 2 is part of continuous queue, and its impact on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = p_2 + p_3 + p_4 - w_4.$$

The new information of job 3 is:

$$\begin{aligned}\Delta w_3 &= r_2 + p_2 + w_2^2 - r_3 - w_3\\ &= p_2 + p_2 + p_3 + p_4 - w_4 + r_2 - r_3 - w_3\\ &= p_2 + p_3 + p_4 - w_4.\\ w_3^2 &= w_3 + \Delta w_3\\ &= p_2 + p_3 + p_4 - w_4 + w_3 > 0.\end{aligned}$$

Now, job 3 belongs to the continuous queue, and its influence on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = p_2 + p_3 + p_4 - w_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= r_3 + p_2 + p_3 + p_4 - w_4 + w_3 + p_3 - r_5 - (r_3 + w_3 + p_3 + p_4 - r_5) \\ &= p_2 + p_3 - w_4.\end{aligned}$$

Finally, the change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_2 + p_3 + p_4 - w_4 + p_2 + p_3 + p_4 - w_4 - w_4 + \pi_5(p_2 + p_3 - w_4) \\ &= p_4 - w_4 + p_2 + p_3 + p_4 - w_4 + p_2 + p_3 - w_4 + \pi_5(p_2 + p_3 - w_4).\end{aligned}$$

2) When $p_2 + p_3 + p_4 - w_4 \leq 0$, job 2 is the queue leader, and its effect on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = 0.$$

The updated information about job 3 is:

$$\begin{aligned}\Delta w_3 &= r_2^o + p_2 - r_3 - w_3 = 0, \\ w_3^2 &= w_3 > 0\end{aligned}$$

For $w_3^2 > 0$, the impact of job 3 on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = 0.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3 + p_3 - r_3 - w_5 \\ &= r_3 + w_3 + p_3 - r_5 - (r_3 + w_3 + p_3 + p_4 - r_5) \\ &= -p_4.\end{aligned}$$

The adjustment of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= -w_4 + \pi_5(-p_4) \\ &= p_4 - w_4 - p_4 + \pi_5(-p_4).\end{aligned}$$

Case 7: $w_4 > 0, w_2 > 0, w_3 \leq 0$

Now, job 3 is the queue leader, both jobs 2 and 4 are in the continuous queue. The change of job 4's waiting time and its new waiting time are respectively:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\ &= r_0^o + p_0 - r_4 - (r_0^o + p_0 + p_2 - w_3 + p_3 - r_4) \\ &= w_3 - p_2 - p_3. \\ w_4^2 &= w_4 + \Delta w_4 \\ &= w_4 + w_3 - p_2 - p_3.\end{aligned}$$

(1) When $w_4 + w_3 - p_2 - p_3 > 0$, job 4 is part of continuous queue, and the adjustment of total waiting time due to job 4 is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = w_3 - p_2 - p_3.$$

The updated information about job 2 is:

$$\begin{aligned}\Delta w_2 &= r_4 + w_4^2 + p_4 - r_2 - w_2 \\ &= w_4 + w_3 - p_2 - p_3 + r_4 + p_4 - r_2 - w_2\end{aligned}$$

$$
\begin{aligned}
&= w_4 + w_3 - p_2 - p_3 + r_2 + p_2 + w_2 - w_3 + p_3 - w_4 + p_4 - r_2 - w_2 \\
&= p_4. \\
w_2^2 &= w_2 + \Delta w_2 \\
&= p_4 + w_2 > 0.
\end{aligned}
$$

For $w_2^2 > 0$, job 2 belongs to the continuous queue, the influence of job 2 on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = p_4.$$

The updated information of job 3 is:

$$
\begin{aligned}
\Delta w_3 &= r_2 + w_2^2 + p_2 - r_3 - w_3 \\
&= r_2 + w_2 + p_4 + p_2 - r_3 - w_3 \\
&= p_4. \\
w_3^2 &= w_3 + \Delta w_3 \\
&= p_4 + w_3.
\end{aligned}
$$

1) If $p_4 + w_3 \leq 0$, at this point, job 3 is the queue leader, and its impact on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = 0.$$

The change of waiting time of job 5 is:

$$\Delta w_5 = r_3^o + p_3 - r_5 - w_5 = -p_4.$$

Finally, the change of total waiting time is:

$$
\begin{aligned}
\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\
&= w_3 - p_2 - p_3 + p_4 + p_4 - p_4 + \pi_5(-p_4)
\end{aligned}
$$

2) If $p_4 + w_3 > 0$, job 3 is in the continuous queue, the impact of job 3 on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = w_3 + p_4.$$

The change of job 5's waiting time is:

$$
\begin{aligned}
w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\
&= p_4 + w_3 - p_4 \\
&= w_3.
\end{aligned}
$$

The change of total waiting time is as follows:

$$
\begin{aligned}
\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\
&= p_4 + w_3 - p_2 - p_3 + p_4 + w_3 + \pi_5(w_3).
\end{aligned}
$$

(2) When $w_4 + w_3 - p_2 - p_3 \leq\leq 0$, job 4 is the queue leader, and its impact on the total waiting time is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = -w_4.$$

The updated waiting time is :

$$
\begin{aligned}
\Delta w_2 &= r_4^o + p_4 - r_2 - w_2 \\
&= (r_2 + p_2 + w_2 - w_3 + p_3 - w_4) + p_4 - r_2 - w_2 \\
&= p_2 + p_3 + p_4 - w_3 - w_4. \\
w_2^2 &= w_2 + \Delta w_2
\end{aligned}
$$

$$= p_2 + p_3 + p_4 - w_3 - w_4 + w_2$$

Because $w_2 > 0$ and $p_2 + p_3 > w_3 + w_4$,we can get that $w_2^2 > 0$. Thus, job 2 is in the continuous queue,and the change of total waiting time caused by job 2 is:

$$\Delta_2 = \Delta w_3^+ = p_2 + p_3 + p_4 - w_3 - w_4.$$

The new information of job 3, once job 4 is moved, is as follows:

$$\begin{aligned}\Delta w_3 &= r_2 + w_2^2 + p_2 - r_3 - w_3 \\ &= r_2 + p_2 + p_3 + p_4 - w_3 - w_4 + w_2 + p_2 - r_3 - (r_2 + w_3 + p_2 - r_3) \\ &= p_2 + p_3 + p_4 - w_3 - w_4. \\ w_3^2 &= w_3 + \Delta w_3 \\ &= p_2 + p_3 + p_4 - w_4.\end{aligned}$$

1) If $p_2 + p_3 + p_4 - w_4 > 0$, job 3 is one of the members of the continuous queue, and its influence on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = p_2 + p_3 + p_4 - w_4 - w_3.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= r_3 + p_2 + p_3 + p_4 - w_4 + p_3 - r_5 - (r_3^o + p_3 + p_4 - r_5) \\ &= p_2 + p_3 - w_4.\end{aligned}$$

Finally, the adjustment of total waiting time is as follows:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= 2p_4 - 2w_4 + p_2 + p_3 - w_3 + p_2 + p_3 - w_4 + \pi_5(p_2 + p_3 - w_4).\end{aligned}$$

2) If $p_2 + p_3 + p_4 - w_4 \leq 0$, job 3 is the queue leader, and its impact on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = 0.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3^o + p_3 - r_5 - w_5 \\ &= r_3^o + p_3 - r_5 - (r_3^o + p_3 + p_4 - r_5) \\ &= -p_4.\end{aligned}$$

Finally, the adjustment of total waiting time caused by job 4's relocation is calculated as:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 - w_4 + p_2 + p_3 + p_4 - w_3 - w_4 - p_4 + \pi_5(-p_4).\end{aligned}$$

Case 8: $w_4 > 0, w_2 > 0, w_3 > 0$

When not moved, job 2, 3 and 4 are belong to the continuous queue. The updated information about job 4, after its relocation, we can get:

$$\begin{aligned}\Delta w_4 &= r_0^o + p_0 - r_4 - w_4 \\ &= r_0^o + p_0 - r_4 - (r_0^o + p_0 + p_2 + p_3 - r_4) \\ &= -p_2 - p_3. \\ w_4^2 &= w_4 + \Delta w_4 \\ &= w_4 - p_2 - p_3.\end{aligned}$$

(1) If $w_4 - p_2 - p_3 > 0$, it means that job 4 is in the continuous queue, and the affect of job 4 on the total waiting time is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = -p_2 - p_3.$$

The new information of job 2 is:

$$\begin{aligned}\Delta w_2 &= r_4 + w_4^2 + p_4 - r_2 - w_2 \\ &= r_4 + w_4 - p_2 - p_3 + p_4 - r_2 - w_2 \\ &= p_4. \\ w_2^2 &= w_2 + \Delta w_2 \\ &= w_2 + p_4.\end{aligned}$$

For $w_2 > 0$, we can get $w_4^2 > 0$. Job 2 in the continuous queue, the impact of job 2 on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = p_4.$$

The change of job 3's waiting time and its new waiting time are, respectively:

$$\begin{aligned}\Delta w_3 &= r_2 + w_2^2 + p_2 - r_3 - w_3 \\ &= r_2 + w_2 + p_4 + p_2 - r_3 - w_3 \\ &= p_4. \\ w_3^2 &= w_3 + \Delta w_3 \\ &= w_3 + p_4.\end{aligned}$$

For $w_3 > 0$, so $w_3^2 > 0$, job 3 is in the continuous queue, the affect of job 3 on the total waiting time is:

$$\Delta_3 = \Delta w_4^+ = p_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= r_3 + w_3 + p_4 + p_3 - r_5 - w_5 \\ &= 0.\end{aligned}$$

The change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 + p_4 - p_2 - p_3 + \pi_5(0).\end{aligned}$$

(2) If $w_4 - p_2 - p_3 \leq 0$, job 4 is the queue leader, and its influence on the total waiting time is:

$$\Delta_4 = \max(0, w_4^2) - \max(0, w_4) = -w_4.$$

The updated information of job 2 is:

$$\begin{aligned}\Delta w_2 &= r_4 + p_4 - r_2 - w_2 \\ &= p_2 + p_3 + p_4 - w_4. \\ w_2^2 &= w_2 + \Delta w_2 \\ &= p_2 + p_3 + p_4 - w_4 + w_2.\end{aligned}$$

For $w_4 \leq p_2 - p_3$ and $w_2 > 0$, so $w_2^2 > 0$. At this point, job 2 is in the continuous queue, and its impact on the total waiting time is:

$$\Delta_2 = \Delta w_3^+ = p_2 + p_3 + p_4 - w_4.$$

The new information of job 3 is:

$$\begin{aligned}\Delta w_3 &= r_2 + w_2^2 + p_2 - r_3 - w_3 \\ &= r_2 + p_2 + p_3 + p_4 - w_4 + w_2 + p_2 - r_3 - w_3 \\ &= p_2 + p_3 + p_4 - w_4. \\ w_3^2 &= w_3 + \Delta w_3 \\ &= p_2 + p_3 + p_4 - w_4 + w_3 > 0.\end{aligned}$$

Similar to job 2, job 3 belongs to the continuous queue, and the total waiting time change due to job 3 is:

$$\Delta_3 = \Delta w_4^+ = p_2 + p_3 + p_4 - w_4.$$

The change of job 5's waiting time is:

$$\begin{aligned}\Delta w_5 &= r_3 + w_3^2 + p_3 - r_5 - w_5 \\ &= r_3 + p_2 + p_3 + p_4 - w_4 + w_3 + p_3 - r_5 - w_5 \\ &= p_2 + p_3 - w_4.\end{aligned}$$

Finally, the change of total waiting time is:

$$\begin{aligned}\Delta w &= \Delta_2 + \Delta_3 + \Delta_4 + \pi_5(\Delta w_5) \\ &= p_4 - w_4 + p_4 + p_2 + p_3 - w_4 + p_2 + p_3 - w_4 + \pi_5(p_2 + p_3 - w_4).\end{aligned}$$

Now, let us integrate the conditions for I_4. I_4 can be expressed as:

$$I_4 = \max(-p_4, \min(0, w_2) + \min(0, w_3), p_2 + p_3 - \max(0, w_4))$$

In the next step, we get the formula of I_3 is:

$$\begin{aligned}I_3 = &\min(0, w_2) + \min(0, w_3) - p_2 - p_3 \\ &+ \max(0, \min(0, w_2) + p_4, p_2 + p_3 + p_4 - \min(0, w_3) - w_4) \\ &+ \max(p_4, p_2 + p_3 + p_4 - \min(0, w_2) - \min(0, w_3) - w_4).\end{aligned}$$

Currently, only the information for w_4^2 is miss, we summarize the w_4^2 as:

$$w_4^2 = \min(0, w_2) + \min(0, w_3) - p_2 - p_3 + w_4.$$

Conclusion: By summarizing all the above information, we get the law that governing the change of queue when job 4 is relocated the position between jobs 1 and 2. The change of total waiting follows this formula:

$$\Delta w = I_3 + I_4 + \pi_5(I_4).$$

where

$$\begin{aligned}I_3 = &\min(0, w_2) + \min(0, w_3) - p_2 - p_3 \\ &+ \max(0, \min(0, w_2) + p_4, p_2 + p_3 + p_4 - \min(0, w_3) - w_4) \\ &+ \max(p_4, p_2 + p_3 + p_4 - \min(0, w_2) - \min(0, w_3) - w_4).\end{aligned}$$

$$I_4 = \max(-p_4, \min(0, w_2) + \min(0, w_3), p_2 + p_3 - \max(0, w_4)).$$

It is obviously that $\max(0, \min(0, w_2) + p_4, p_2 + p_3 + p_4 - \min(0, w_3) - w_4)$ and $\max(p_4, p_2 + p_3 + p_4 - \min(0, w_2) - \min(0, w_3) - w_4)$ are in line with recursive formula of Lemma 2. So, the equivalent formula of I_3 can be written as:

$$I_3 = \min(0, w_2) + \min(0, w_3) - p_2 - p_3 + \sum_{j=2}^{3} \Delta w_j^+$$

$$= \sum_{j=2}^{3}(\min(0, w_j) - p_j) + \sum_{j=2}^{3} \Delta w_j^+.$$

where the initial value is $\Delta w_2^+ = \max(p_4, p_2 + p_3 + p_4 - \min(0, w_2) - \min(0, w_3) - w_4)$. The new waiting time of w_4^2 can be calculated as:

$$w_4^2 = \min(0, w_2) + \min(0, w_3) - p_2 - p_3 + w_4.$$

This law can be applied to the backward movement of job $\sigma_t(i)$ in any sequence σ_t to the position before $\sigma_t(k)$(where $k \in [1, i-1]$). The change of total waiting time can be expressed as:

$$\Delta w = I_3 + I_4 + \pi_{\sigma_t(k+1)}(I_4).$$

where

$$I_3 = \sum_{j=k}^{i-1}(\min(0, w_{\sigma_t(j)}) - p_{\sigma_t(j)}) + \sum_{j=k}^{i-1} \Delta w_{\sigma_t(j)}^+.$$

$$I_4 = \max\left(-p_{\sigma_t(i)}, \sum_{j=k}^{i-1} \min(0, w_{\sigma_t(j)}), \sum_{j=k}^{i-1} p_{\sigma_t(j)} - \max(0, w_{\sigma_t(i)})\right).$$

where the initial value of the recursive formula is:

$$\Delta w_{\sigma_t(k)}^+ = \max\left(p_{\sigma_t(i)}, \sum_{j=k}^{i} p_{\sigma_t(j)} - \sum_{j=k}^{i-1} \min(0, w_{\sigma_t(j)}) - w_{\sigma_t(i)}\right).$$

The job $\sigma_t(i)$'s new waiting time $w_{\sigma_t(i)}^2$ is calculated as:

$$w_{\sigma_t(i)}^2 = \sum_{j=k}^{i-1}(\min(0, w_{\sigma_t(j)}) - p_{\sigma_t(j)}) + w_{\sigma_t(i)}.$$

□

A.6 Proof of Proposition 2

Proof The proposition's first inequality and second inequality outline the relationships that waiting time and processing time are required to meet for the first-come, first-served rule and the first-come, last-served rule, respectively.

For two adjacent jobs $\sigma_t(i)$ and $\sigma_t(i+1)$ in the queue σ_t. When $r_{\sigma_t(i)} \le r_{\sigma_t(i+1)}$, it reflects the first-come, first-served rule. In that case, the maximum waiting time for $\sigma_t(i+1)$ is the waiting time of $\sigma_t(i)$ plus its processing time. A greater waiting time could only occur if it arrives prior to $\sigma_t(i)$ but is serviced after $\sigma_t(i)$, which corresponds to a first-come, last-served rule.

To simplify the derivation, let $i = \sigma_t(i)$ and $i+1 = \sigma_t(i+1)$, with the service sequence always having job i precede $i+1$. The release time of the previous queue leader and the cumulative processing time up to job i are r_0^o and p_0^o, respectively. As defined earlier, the waiting time for job i is

$$w_i = r_0^o + p_0^o - r_i.$$

The waiting time calculation for job $i+1$ needs to account for changes of the queue leader:

- When $w_i > 0, w_{i+1} = r_0^o + p_0^o + p_i - r_{i+1}$.
- When $w_i \le 0, w_{i+1} = r_i^o + p_i - r_{i+1}$.

The two inequalities of the proposition that require proof are as follows:

- When $r_i \leq r_{i+1}, \max(0, w_i) + p_i \geq \max(0, w_{i+1})$.
- When $r_i > r_{i+1}, \max(0, w_i) + p_i < \max(0, w_{i+1})$.

As can be observed, the positivity or negativity of w_i and w_{i+1} is closely related to the proof of the inequalities. We will classify cases according to them.

Firstly, we prove the inequality for the first part under the first-come, first-served principle($r_{i+1} \geq r_i$):

Case 1: $w_i > 0, w_{i+1} > 0$, both job i and $i+1$ are in the same continuous queue. We have

$$\max(0, w_i) + p_i = w_i + p_i = r_0^o + p_0^o - r_i + p_i,$$

$$\max(0, w_{i+1}) = w_{i+1} = r_0^o + p_0^o + p_i - r_{i+1},$$

$$\because \text{the first-come, first-served } r_{i+1} \geq r_i, \therefore \max(0, w_i) + p_i \geq \max(0, w_(i+1)).$$

The inequality holds.

Case 2: $w_i > 0, w_{i+1} \leq 0$, job i remains in the continuous queue, while job $i+1$ is the queue leader. At this point, we get

$$\begin{aligned} &\max(0, w_i) + p_i \\ &= w_i + p_i \\ &= r_0^o + p_0^o - r_i + p_i \geq 0 = \max(0, w_{i+1}) \end{aligned}$$

The inequality can be confirmed.

Case 3: $w_i \leq 0, w_{i+1} > 0$, job i is the queue leader, while job $i+1$ belongs to the continuous queue. So we have

$$\max(0, w_i) + p_i = p_i,$$

$$\max(0, w_{i+1}) = w_{i+1} = r_i^o + p_i - r_{i+1},$$

$$\because r_{i+1} \geq r_i^o, \therefore p_i \geq r_i^o + p_i - r_{i+1},$$

$$\therefore \max(0, w_i) + p_i \geq \max(0, w_{i+1}).$$

The inequality is shown to hold.

Case 4: $w_i \leq 0, w_{i+1} \leq 0$, both jobs are the queue leader. At this time, we get

$$\max(0, w_i) + p_i = p_i \geq 0 = \max(0, w_{i+1}).$$

Obviously, the inequality continues to hold.

As discussed above, the first inequality of the proposition is verified. We will explore the second portion of the proposition under the first-served, last-come rule. Under this rule, when the release time r_i of job i is later than that of job $i+1$, and the service order of job i precedes that of job $i+1$, the waiting time w_{i+1} of job $i+1$ is bound to be greater than 0. Therefore, only two cases need to be discussed.

Case 1: $w_i > 0, w_{i+1} > 0$, both job i and $i+1$ are in the same continuous queue. We have

$$\max(0, w_i) + p_i = w_i + p_i = r_0^o + p_0^o - r_i + p_i,$$

$$\max(0, w_{i+1}) = w_{i+1} = r_0^o + p_0^o + p_i - r_{i+1},$$

$$\because \text{the first-come, last-served } r_{i+1} < r_i, \therefore \max(0, w_i) + p_i < \max(0, w_(i+1)).$$

The inequality holds.

Case 2: $w_i \leq 0, w_{i+1} > 0$, job i is the queue leader, while job $i+1$ belongs to the continuous queue. So we have

$$\max(0, w_i) + p_i = p_i,$$

$$\max(0, w_{i+1}) = w_{i+1} = r_i^o + p_i - r_{i+1},$$

$$\because \text{ the first-come, last-served } r_{i+1} < r_i^o, \therefore p_i < r_i^o + p_i - r_{i+1},$$
$$\therefore \max(0, w_i) + p_i < \max(0, w_{i+1}).$$

The inequality is proved.

So far, we have validated the relationships between the waiting time and processing time of adjacent jobs required under the first-come, first-served, and first-come, last-served principles, respectively, and Proposition 1 is proven. □

B Omitted Proofs in Section 3

B.1 Proof of Lemma 3

Proof Theorem 1 provides the formula for the variation in the total waiting time caused by a forward move of a job. Any forward move yielding $\Delta w \leq 0$ is regarded as an improving solution. Once the expression for Δw has been derived, it can be further analyzed according to the two key terms I_1 and I_2. This leads to the following four mutually exclusive cases:

Case 1: $I_1 \leq 0,\ I_2 \geq 0$;

Case 2: $I_1 > 0,\ I_2 > 0$;

Case 3: $I_1 \leq 0,\ I_2 \leq 0$;

Case 4: $I_1 > 0,\ I_2 < 0$.

Consider an arbitrary schedule σ_t and any four consecutive jobs $\sigma_t(i)$, $\sigma_t(i+1)$, and $\sigma_t(i+2)$, where $i \in \{1, \ldots, n-2\}$. Without loss of generality, relabel them as $4, 5, 6$, respectively. We analyze the move in which job 4 is moved forward to the position immediately after job 6.

Case 2 can never reduce the total waiting time and can therefore be excluded directly. Case 4 requires $I_2 < 0$. A necessary condition for $I_2 < 0$ is

$$w_4 < 0, \quad w_5 > 0,\ w_6 > 0, \quad p_4 - w_5 < 0,\ p_4 - w_6 < 0.$$

Case 2 is clearly non-improving and can be excluded immediately. For case 4, the condition $I_2 < 0$ is required, which in turn implies the necessary conditions

$$w_4 < 0, \qquad w_5 > 0,\ w_6 > 0, \qquad p_4 - w_5 < 0,\ p_4 - w_6 < 0.$$

If the original sequence follows the FCFS order, then $r_4 \leq r_5$. By Proposition 2,

$$\max(0, w_4) + p_4 \geq \max(0, w_5).$$

Since $w_4 < 0$ and $w_5 > 0$, this reduces to

$$p_4 - w_5 \geq 0,$$

contradicting $p_4 - w_5 < 0$. Hence, under the FCFS order, case (iv) cannot occur.

If, after some adjustments, the release-time order is reversed so that $r_4 > r_5$, then Proposition 2 yields

$$\max(0, w_4) + p_4 < \max(0, w_5),$$

and thus $p_4 < w_5$. In this situation, the queue is discontinuous at job 4 because $w_4 < 0$. Such a case is handled by the Bottleneck Breakthrough Rule, and therefore need not be considered in the present analysis of forward candidate solutions.

Cases 1 and 3 can be unified into the single situation $I_1 \leq 0$. By Lemma 1, the upper bound of Δw_5^- is determined by its initial value:

$$\Delta w_5^- \leq p_4 - \min(0, w_4).$$

Consider the most favorable upper bound for the propagation, namely, the extreme case in which $\Delta w_{\sigma_t(j)}^- = p_4 - \min(w_4, 0)$ for all relevant j, corresponding to a situation where the decrease flow does not attenuate during propagation. This yields the most conservative estimate of I_1:

$$\begin{aligned}I_1 &= \sum_{j=5}^{6}\Big(p_{\sigma_t(j)} - \min(0, w_{\sigma_t(j)})\Big) - \sum_{j=6}^{7}\Delta w^-_{\sigma_t(j)}\\ &\geq \sum_{j=5}^{6} p_{\sigma_t(j)} - \sum_{j=6}^{7}\Delta w^-_{\sigma_t(j)}.\end{aligned}$$

Combining $\Delta w_5^- \leq p_4 - \min(0, w_4)$ with the monotonicity in Lemma 1, we further obtain $\Delta w_7^- \leq \Delta w_5^-$. Therefore,

$$\begin{aligned}I_1 &\geq \sum_{j=5}^{6} p_{\sigma_t(j)} - \sum_{j=6}^{7}\Delta w^-_{\sigma_t(j)}\\ &\geq \sum_{j=5}^{6}\Big(p_{\sigma_t(j)} - \Delta w^-_{\sigma_t(j)}\Big)\\ &\geq \sum_{j=5}^{6}\Big(p_{\sigma_t(j)} - \big(p_4 - \min(0, w_4)\big)\Big).\end{aligned}$$

It follows that this condition is necessary for a forward move to reduce the total waiting time. If job $\sigma_t(i)$ is moved forward to position k and satisfies

$$\sum_{j=i+1}^{k}\Big(p_{\sigma_t(j)} - \big(p_{\sigma_t(i)} - \min(w_0, \sigma_t(i))\big)\Big) \leq 0,$$

then such a move may reduce the total waiting time. We call position k a *forward candidate solution* of job $\sigma_t(i)$, denoted by $\theta^f_{i,k}$. □

B.2 Proof of Lemma 4

Proof To complete the structural characterization of the solution space, we further analyze the improvement mechanism induced by a backward move of a job. Consider an arbitrary sequence σ_t, and select any five consecutive jobs

$$\sigma_t(i),\ \sigma_t(i+1),\ \sigma_t(i+2),\ \sigma_t(i+3),\ \sigma_t(i+4), \qquad i \in \{1, \ldots, n-4\}.$$

For notational simplicity and without loss of generality, let

$$\sigma_t(i) = 2, \quad \sigma_t(i+1) = 3, \quad \sigma_t(i+2) = 4, \quad \sigma_t(i+3) = 5, \quad \sigma_t(i+4) = 6.$$

This relabeling is introduced only for local analysis and does not affect the general validity of the argument.

Now consider moving job 4 backward to the position immediately before job 2. After this move, the waiting times of jobs 2 and 3 may increase, whereas job 4 is processed earlier and its waiting time may decrease. Meanwhile, the waiting-time variations of job 5 and all subsequent jobs are determined by the propagation effect. Let $\Delta_4 \geq 0$ denote the improvement contributed by job 4, that is, the change in its contribution to the objective value is $-\Delta_4$. Let $\pi_5(\Delta w_5)$ denote the propagation effect starting from job 5. Then the total variation in the total waiting time caused by this backward move can be written as

$$\Delta w = \Delta_2 + \Delta_3 + \pi_5(\Delta w_5) - \Delta_4,$$

where $\Delta_2, \Delta_3 \geq 0$ represent the nonnegative deterioration contributions of jobs 2 and 3, respectively.

We first show that

$$-\Delta_4 + \Delta w_5 \leq 0$$

is a necessary condition for this backward move to improve the total waiting time.

Step 1: Proof of the necessary condition.

If $\Delta w_5 \leq 0$, then by the propagation rule in Lemma 2, the propagation contribution generated by job 5 and all subsequent jobs satisfies

$$\pi_5(\Delta w_5) \leq 0.$$

In this case, the downstream jobs cannot offset the improvement brought by job 4, and therefore

$$-\Delta_4 + \Delta w_5 \leq 0$$

cannot be violated. Hence, the necessary condition holds trivially in this case.

We next consider the case $\Delta w_5 > 0$. In this case, the propagation effect starting from job 5 becomes a nonnegative deterioration term. By Lemma 2,

$$\Delta_3 \geq 0, \qquad \pi_5(\Delta w_5) \geq 0,$$

and thus

$$\Delta w = \Delta_2 + \Delta_3 + \pi_5(\Delta w_5) - \Delta_4 \geq \Delta_2 - \Delta_4.$$

It therefore remains to derive a lower bound for $\Delta_2 - \Delta_4$.

By Theorem 2, when $\Delta w_5 > 0$,

$$\Delta w_5 = I_4 = p_2 + p_3 - \max(0, w_4) > 0.$$

Hence,

$$p_2 + p_3 > \max(0, w_4) \geq w_4.$$

The increase in the waiting time of job 2 is

$$\Delta w_2^+ = \max\left\{ p_4,\ \sum_{j=2}^{4} p_j - \sum_{j=2}^{3} \min(0, w_j) - w_4 \right\}.$$

Since

$$\sum_{j=2}^{4} p_j - \sum_{j=2}^{3} \min(0, w_j) - w_4 \geq p_4 + \big(p_2 + p_3 - w_4\big) \geq p_4,$$

the maximum is attained by the second term, and therefore

$$\Delta w_2^+ = \sum_{j=2}^{4} p_j - \sum_{j=2}^{3} \min(0, w_j) - w_4.$$

Accordingly, the contribution of job 2 to the total waiting time is

$$\begin{aligned} \Delta_2 &= \max\big(0, \Delta w_2^+ + \min(0, w_2)\big) \\ &= \sum_{j=2}^{4} p_j - \min(0, w_3) - w_4. \end{aligned}$$

On the other hand, the new waiting time of job 4 after the move is

$$w_4^2 = w_4 + \min(0, w_2) + \min(0, w_3) - p_2 - p_3.$$

From

$$p_2 + p_3 - \max(0, w_4) > 0$$

it follows that

$$p_2 + p_3 - w_4 > 0.$$

Combining this with

$$-\min(0, w_2) \geq 0, \qquad -\min(0, w_3) \geq 0,$$

we obtain

$$p_2 + p_3 - \min(0, w_2) - \min(0, w_3) - w_4 > 0,$$

that is,

$$w_4^2 < 0.$$

Therefore, the waiting time of job 4 after the move remains nonpositive, and its improvement in the objective value is

$$-\Delta_4 = -\max(0, w_4).$$

Hence,

$$\begin{aligned}
\Delta_2 - \Delta_4 &= \max\bigl(0, \Delta w_2^+ + \min(0, w_2)\bigr) - \max(0, w_4) \\
&= \sum_{j=2}^{4} p_j - \min(0, w_3) - w_4 - \max(0, w_4) \\
&\geq p_2 + p_3 - \max(0, w_4) - \max(0, w_4).
\end{aligned}$$

Since

$$\Delta w_5 = p_2 + p_3 - \max(0, w_4),$$

we have

$$\Delta_2 - \Delta_4 \geq -\Delta_4 + \Delta w_5.$$

Therefore,

$$\Delta w \geq \Delta_2 - \Delta_4 \geq -\Delta_4 + \Delta w_5.$$

Consequently, if

$$-\Delta_4 + \Delta w_5 > 0,$$

then necessarily $\Delta w > 0$, and the backward move cannot reduce the total waiting time. The contrapositive statement is therefore: if a backward move satisfies $\Delta w \leq 0$, then it must hold that

$$-\Delta_4 + \Delta w_5 \leq 0.$$

Hence, this inequality is a necessary condition for a backward move to improve the total waiting time.

Step 2: Construction of the discriminant function and boundary analysis.
Define

$$H(p_4, w_4) := -\Delta_4 + \Delta w_5.$$

By Step 1, $H(p_4, w_4) \leq 0$ is a necessary condition for the backward move to be potentially improving. We next examine the structure of H according to the value of w_4.

Case 1: $w_4 \leq 0$.
By Theorem 2,

$$\begin{aligned}
w_4^2 &= \min(0, w_3) + \min(0, w_2) - p_3 - p_2 + w_4, \\
\Delta w_5 = I_4 &= \max\bigl\{-p_4,\ \min(0, w_2) + \min(0, w_3),\ p_2 + p_3 - \max(0, w_4)\bigr\}.
\end{aligned}$$

When $w_4 \leq 0$, we have $\max(0, w_4) = 0$, so the third term becomes $p_2 + p_3 \geq 0$, whereas the first two terms are both no greater than 0. Thus,

$$\Delta w_5 = p_2 + p_3.$$

Moreover, since $w_4 \le 0$, it follows directly that $w_4^2 \le 0$, and hence

$$-\Delta_4 = 0.$$

Therefore,

$$H(p_4, w_4) = p_2 + p_3 \ge 0.$$

Thus, when $w_4 \le 0$, this backward move can never improve the total waiting time.

Case 2: $w_4 > 0$.
In this case,

$$\begin{aligned} w_4^2 &= \min(0, w_3) + \min(0, w_2) - p_3 - p_2 + w_4, \\ \Delta w_5 = I_4 &= \max\{-p_4,\ \min(0, w_2) + \min(0, w_3),\ p_2 + p_3 - w_4\}. \end{aligned}$$

We further divide the analysis into two subcases.

Subcase 2.1: $w_4^2 > 0$.
In this case,

$$\min(0, w_2) + \min(0, w_3) - p_2 - p_3 + w_4 > 0,$$

and therefore the change in the objective contribution of job 4 is

$$-\Delta_4 = \min(0, w_2) + \min(0, w_3) - p_2 - p_3.$$

Moreover, since

$$\min(0, w_2) + \min(0, w_3) > p_2 + p_3 - w_4,$$

we have

$$\Delta w_5 = \max\{-p_4,\ \min(0, w_2) + \min(0, w_3)\}.$$

If

$$\min(0, w_2) + \min(0, w_3) > -p_4,$$

then

$$H(p_4, w_4) = 2\min(0, w_2) + 2\min(0, w_3) - p_2 - p_3 \le 0.$$

If

$$\min(0, w_2) + \min(0, w_3) \le -p_4,$$

then

$$H(p_4, w_4) = \min(0, w_2) + \min(0, w_3) - p_2 - p_3 - p_4 \le 0.$$

Subcase 2.2: $w_4^2 \le 0$.
In this case,

$$\min(0, w_2) + \min(0, w_3) \le p_2 + p_3 - w_4,$$

and thus

$$-\Delta_4 = -w_4, \qquad \Delta w_5 = \max\{-p_4,\ p_2 + p_3 - w_4\}.$$

If

$$p_2 + p_3 - w_4 > -p_4,$$

then

$$H(p_4, w_4) = p_2 + p_3 - 2w_4.$$

Therefore, when

$$w_4 \ge \frac{p_2 + p_3}{2},$$

it holds that $H(p_4, w_4) \le 0$.
If

$$p_2 + p_3 - w_4 \le -p_4,$$

then

$$H(p_4, w_4) = -w_4 - p_4 \leq 0.$$

In summary, in the case $w_4 > 0$, the condition $H(p_4, w_4) \leq 0$ characterizes the possibility that the backward move may still improve the total waiting time.

Step 3: Extension from the local structure to a general backward candidate.
Extending the local pair $(2, 3)$ in the above analysis to a general interval $(j, \ldots, i-1)$ yields the discriminant condition for a general backward move.

Furthermore, the above piecewise representation can be unified as

$$\begin{aligned} H(p_4, w_4) = & \max\bigl(-w_4,\ \min(0, w_2) + \min(0, w_3),\ -p_2 - p_3\bigr) \\ & + \max\bigl(-p_4,\ p_2 + p_3 - w_4,\ \min(0, w_2) + \min(0, w_3)\bigr). \end{aligned}$$

Hence, H is the sum of maxima of finitely many affine functions, and is therefore a piecewise-linear convex function. Its minimum is attained at the boundaries where the dominant affine pieces switch. Accordingly, for any position $j \in [1, i-1]$, if

$$\begin{aligned} w_{\sigma_t(i)} &= \sum_{k=j}^{i-1} \bigl(p_{\sigma_t(k)} - \min(0, w_{\sigma_t(k)})\bigr), \\ p_{\sigma_t(i)} &= \sum_{k=j}^{i-1} \bigl(-\min(0, w_{\sigma_t(k)})\bigr), \end{aligned}$$

then, by the above analysis, equality corresponds exactly to the critical boundary at which the backward move reaches the threshold and the function H attains its minimum. Moreover, only when the corresponding cumulative quantities are below this boundary can the waiting time of the moved job remain nonnegative after the move. If the waiting time of the moved job remains negative after the move, then any further backward move cannot improve the total waiting time. Therefore, for a backward candidate to be meaningful in practice, one must further require

$$w_{\sigma_t(i)} \geq \sum_{k=j}^{i-1} \bigl(p_{\sigma_t(k)} - \min(0, w_{\sigma_t(k)})\bigr),\ \wedge\ p_{\sigma_t(i)} \geq \sum_{k=j}^{i-1} \bigl(-\min(0, w_{\sigma_t(k)})\bigr).$$

Any position j satisfying the above inequalities is called a backward candidate position of job $\sigma_t(i)$, denoted by $\theta^b_{i,j}$.

This condition indicates that only when job $\sigma_t(i)$ satisfies the above inequalities can inserting $\sigma_t(i)$ backward into position j potentially reduce the objective value. Therefore, the above two inequalities provide the necessary discriminant conditions for the set of backward candidate solutions. □

B.3 Proof of Lemma 7

Proof We first show that only a job located after position i can repair the queue discontinuity at position i. Taking position i as the dividing point, we partition the prefix of the sequence into

$$P_a = \{\sigma_t(1), \ldots, \sigma_t(i-1)\}.$$

Viewing P_a as an aggregate block, its completion time C_{P_a} can be written as

$$C_{P_a} = C_{P_o} + \sum_{j \in P_a} p_j + \sum_{j \in P_a} \bigl(-\min(0, w_j)\bigr),$$

where C_{P_o} is the completion time immediately before P_a, $\sum_{j\in P_a} p_j$ is the total processing time of the jobs in P_a, and $\sum_{j\in P_a}\big(-\min(0,w_j)\big)$ is the total machine idle time accumulated within P_a. Since the set of jobs in P_a is fixed, the total processing time is constant. It follows that any change in C_{P_a} is entirely determined by the total machine idle time within P_a.

To repair the queue discontinuity at job $\sigma_t(i)$, one must increase C_{P_a}, that is, delay the completion time of the entire block P_a. Since the total processing time of P_a remains unchanged, this is equivalent to shifting an equal amount of machine idle time from after position i into P_a, thereby postponing the service start time of job $\sigma_t(i)$. Under such an adjustment, job $\sigma_t(i)$ and all subsequent jobs are affected by an increasing flow. However, for an increasing flow, a queue discontinuity does not obstruct the realization of the ideal move, and therefore no repair is needed there.

Therefore, the region that truly requires repair is the one affected by the decreasing flow. This implies that the repair must be achieved by moving some job located after position i forward, so as to repair the queue discontinuity at position i.

Next, we prove that, among the jobs located after position i, only those whose release times are strictly smaller than that of job $\sigma_t(i)$ can repair the discontinuity.

At this point, we have $w_{\sigma_t(i)} \le 0$. Since Proposition 2 only determines the release-time order of two adjacent jobs, we first move job $\sigma_t(j)$ to position $i+1$, and denote its waiting time there by $w'_{\sigma_t(j)}$. By Theorem 2,

$$w'_{\sigma_t(j)} = w_{\sigma_t(j)} - \sum_{k=i+1}^{j-1} \big(p_{\sigma_t(k)} - \min(0, w_{\sigma_t(k)})\big).$$

Then move job $\sigma_t(j)$ further to position i, and denote its waiting time by $w^2_{\sigma_t(j)}$. Again by Theorem 2,

$$w^2_{\sigma_t(j)} = w_{\sigma_t(j)} - \sum_{k=i}^{j-1} \big(p_{\sigma_t(k)} - \min(0, w_{\sigma_t(k)})\big).$$

Under the standing assumption $w_{\sigma_t(i)} \le 0$, we have

$$w_{\sigma_t(i)} = \min(0, w_{\sigma_t(i)}).$$

If the queue discontinuity at job $\sigma_t(i)$ can be repaired, then it is necessary that

$$w^2_{\sigma_t(j)} > w_{\sigma_t(i)}.$$

Substituting the above expression gives

$$w'_{\sigma_t(j)} - p_{\sigma_t(i)} + w_{\sigma_t(i)} > w_{\sigma_t(i)},$$

that is,

$$w'_{\sigma_t(j)} > p_{\sigma_t(i)}.$$

Since $p_{\sigma_t(i)} \ge 0$, it follows that

$$w'_{\sigma_t(j)} > 0.$$

Therefore, the criterion in Proposition 2 is satisfied:

$$w'_{\sigma_t(j)} = \max(0, w'_{\sigma_t(j)}) > \max(0, w_{\sigma_t(i)}) + p_{\sigma_t(i)} = p_{\sigma_t(i)},$$

where the last equality holds because $w_{\sigma_t(i)} \le 0$. Hence,

$$r_{\sigma_t(j)} < r_{\sigma_t(i)}.$$

We thus conclude that the queue discontinuity at position i can be repaired only if job $\sigma_t(j)$ satisfies

$$r_{\sigma_t(j)} < r_{\sigma_t(i)} \qquad \text{and} \qquad w'_{\sigma_t(j)} > 0.$$

Equivalently, only when the release time of job $\sigma_t(j)$ is strictly smaller than that of job $\sigma_t(i)$, and its waiting time after being moved to position $i+1$ is positive, can the queue discontinuity at job $\sigma_t(i)$ be repaired. $\square$

B.4 Proof of Lemma 8

Proof First, by Lemma 7, any job that can repair discontinuity b must have a release time strictly smaller than that of the job at the discontinuity. Therefore, the jobs capable of repairing discontinuity b are ordered nondecreasingly by release times around the discontinuity.

Next, consider the propagation of the decrease flow induced when a compensation job moves forward. By the recursive formula for the decrease flow of waiting times, the initial decrease-flow value of job $\sigma^t(j)$ is

$$\Delta^-_{\sigma^t(j+1)} = p_{\sigma^t(j)} - \min\bigl(0, w_{\sigma^t(j)}\bigr).$$

The initial value of this theoretical propagation upper bound can be uniformly tightened to the processing time $p_{\sigma^t(j)}$. Specifically, if $w_{\sigma^t(j)} > 0$, then

$$\Delta^-_{\sigma^t(j+1)} = p_{\sigma^t(j)}.$$

If $w_{\sigma^t(j)} \leq 0$, although

$$\Delta^-_{\sigma^t(j+1)} = p_{\sigma^t(j)} - w_{\sigma^t(j)} \geq p_{\sigma^t(j)},$$

it follows from the recursive decrease-flow formula together with the relation in Proposition 2,

$$\max\bigl(0, w_{\sigma^t(j)}\bigr) + p_{\sigma^t(j)} \geq \max\bigl(0, w_{\sigma^t(j+1)}\bigr),$$

that once this initial decrease flow enters the subsequent continuous queue, it is immediately truncated by the waiting times of the subsequent jobs. Hence, the theoretical upper bound on its further propagation may still be uniformly regarded as $p_{\sigma^t(j)}$. Therefore, when constructing the forward candidate set, the initial upper bound of the compensation job's propagation can be tightened to its processing time.

Accordingly, under the nondecreasing order of release times, for any given continuous queue interval C, a necessary condition for the forward candidate set of job $\sigma^t(j)$ can be written as

$$\sum_{l \in C} \bigl(p_{\sigma^t(l)} - p_{\sigma^t(j)}\bigr) \leq 0.$$

Similarly, for any job $\sigma^t(h_q) \in B$, we have

$$\sum_{l \in C} \bigl(p_{\sigma^t(l)} - p_{\sigma^t(h_q)}\bigr) \leq 0.$$

It should be emphasized that these conditions correspond to the forward candidate sets derived from the upper bound of the decrease flow, and are used to characterize the forward range that may be covered by an ideal improvement direction. The actually reachable range is further truncated by the true waiting-time structure and is therefore usually smaller than this theoretical range.

We now compare the covering relation between job $\sigma^t(j)$ and job $\sigma^t(h_q)$ under the same discontinuity b.

Case 1: $p_{\sigma^t(h_q)} < p_{\sigma^t(j)}$. For the same region C, the necessary condition for a theoretical forward candidate can be written uniformly as

$$\sum_{l \in C} \bigl(p_{\sigma^t(l)} - p\bigr) \leq 0,$$

where p denotes the processing time of the job serving as the local compensation benchmark. This condition is equivalent to

$$\sum_{l \in C} p_{\sigma^t(l)} \leq |C|\, p.$$

Therefore, this necessary condition is monotone nondecreasing in the benchmark processing time p: if $p_1 \leq p_2$, then the set of theoretical candidate endpoints corresponding to p_1 must be contained in that corresponding to p_2. Hence, under the same discontinuity and the same family of intervals, a compensation job with a larger processing time has a theoretical forward candidate coverage range no smaller than that of a job with a smaller processing time. Therefore, in this case, the theoretical forward candidate traversal range of $\sigma^t(j)$ covers the current position of job $\sigma^t(h_q)$.

Case (2): $p_{\sigma^t(h_q)} \geq p_{\sigma^t(j)}$. In this case, under the tightened upper bound on the initial decrease flow, at the same position, the decrease flow induced by job $\sigma^t(j)$ is no greater than that induced by job $\sigma^t(h_q)$. However, by the greedy rule, job $\sigma^t(j)$ is selected because its immediate compensation at discontinuity b is no smaller than that of any candidate job. Hence, under the same discontinuity, if jobs $\sigma^t(j)$ and $\sigma^t(h_q)$ are respectively moved to discontinuity b and then moved forward back to their original positions, the total accumulated idle time induced by the former process is no smaller than that induced by the latter.

By the monotone decay property of the recursive formula for the decrease flow (Lemma 1), if job $\sigma^t(j)$, which has weaker initial propagation capability, is to attain a total accumulated idle time no smaller than that of job $\sigma^t(h_q)$, then it can do so only by relying on a propagation length no shorter than that of $\sigma^t(h_q)$. On the other hand, the necessary condition defining the forward candidate set is itself based on accumulated-sum tests over consecutive queue intervals, and hence its candidate endpoints advance continuously along the queue in the forward direction by definition. That is, once the accumulated idle time associated with a job is sufficient to cover a longer consecutive interval, then all positions corresponding to shorter consecutive intervals preceding it must also fall within its forward candidate traversal range.

Therefore, in the present case, as long as the propagation length of the decrease flow induced by job $\sigma^t(j)$ is no shorter than that of job $\sigma^t(h_q)$, the consecutive interval corresponding to the current position of job $\sigma^t(h_q)$ must be contained in the family of forward candidate intervals of job $\sigma^t(j)$. Hence, the forward candidate traversal range of job $\sigma^t(j)$ also covers the current position of job $\sigma^t(h_q)$.

Combining the two cases, for any $\sigma^t(h_q) \in B$, its current position is contained in the traversal range of the forward candidate set induced by job $\sigma^t(j)$. This completes the proof. $\square$

C Omitted Proofs in Section 4

C.1 Proof of Lemma 10

Proof Consider an arbitrary sequence σ_t and any two adjacent jobs $\sigma_t(i)$ and $\sigma_t(i+1)$. For ease of exposition, and without loss of generality, let $\sigma_t(i) = 4$ and $\sigma_t(i+1) = 5$. This relabeling is used only to denote the adjacent pair and has no effect on the subsequent derivation.

Swapping jobs 4 and 5 is equivalent to moving job 4 to the position immediately after job 5. After the swap, the waiting times of the two jobs are given by

$$w_4^2 = \max(0, w_4) + p_4 + p_5 - \min(0, w_4) - \Delta w_6^-,$$
$$w_5^2 = w_5 - \Delta w_5^- = w_5 - \big(p_4 - \min(0, w_4)\big).$$

Let Δw_{pos} denote the change in the sum of the positive waiting times of jobs 4 and 5 before and after the swap. Then

$$\Delta w_{\text{pos}} = \max(0, w_4^2) + \max(0, w_5^2) - \max(0, w_4) - \max(0, w_5).$$

If the swap is to reduce the total waiting time of jobs 4 and 5, it is necessary that

$$\Delta w_{\text{pos}} < 0.$$

Let Δw_{neg} denote the change in machine idle time before and after the swap, namely,

$$\Delta w_{\text{neg}} = \min(0, w_4^2) + \min(0, w_5^2) - \min(0, w_4) - \min(0, w_5).$$

If $\Delta w_{\text{neg}} < 0$, then the waiting times of the subsequent jobs increase; if $\Delta w_{\text{neg}} = 0$, then the subsequent jobs are unaffected; and if $\Delta w_{\text{neg}} > 0$, then the waiting times of the subsequent jobs decrease.

Now suppose that jobs 4 and 5 satisfy the first-come-first-served condition in Proposition 1, namely,

$$\max(0, w_4) + p_4 \geq \max(0, w_5).$$

Case 1: $w_4 \leq 0$.

In this case, from $\max(0, w_4) + p_4 \geq \max(0, w_5)$ it follows that $w_5 \leq p_4$.

Subcase 1.1: $0 < w_5 \leq p_4$.

The new waiting times of jobs 4 and 5 after the swap are

$$\begin{aligned} w_4^2 &= \max(0, w_4) + p_4 + p_5 - \min(0, w_4) - \Delta w_6^- \\ &= p_4 + p_5 - w_5, \\ w_5^2 &= w_5 - \big(p_4 - \min(0, w_4)\big) = w_5 - p_4 + w_4. \end{aligned}$$

Since $0 < w_5 \leq p_4$, we have

$$w_4^2 = p_4 + p_5 - w_5 \geq 0, \qquad w_5^2 = w_5 - p_4 + w_4 \leq 0.$$

The change in machine idle time is

$$\begin{aligned} \Delta w_{\text{neg}} &= \min(0, w_4^2) + \min(0, w_5^2) - \min(0, w_4) - \min(0, w_5) \\ &= w_5 - p_4. \end{aligned}$$

Because $w_5 \leq p_4$, it follows that $\Delta w_{\text{neg}} = w_5 - p_4 \leq 0$. Hence, if $w_5 < p_4$, the waiting times of the subsequent jobs increase; if $w_5 = p_4$, they remain unchanged.

The change in positive waiting time is

$$\begin{aligned} \Delta w_{\text{pos}} &= \max(0, w_4^2) + \max(0, w_5^2) - \max(0, w_4) - \max(0, w_5) \\ &= p_4 + p_5 - w_5 - w_5 \\ &= p_4 + p_5 - 2w_5. \end{aligned}$$

Therefore,

$$\Delta w_{\text{pos}} < 0 \iff p_4 + p_5 - 2w_5 < 0 \iff w_5 > \frac{1}{2}(p_4 + p_5).$$

Thus, $w_5 > \frac{1}{2}(p_4+p_5)$ is the necessary and sufficient condition for $\Delta w_{\text{pos}} < 0$ in this subcase.

Accordingly, under $0 < w_5 \leq p_4$, the condition $\Delta w_{\text{pos}} < 0$ is equivalent to

$$w_5 > \frac{1}{2}(p_4 + p_5).$$

Moreover, combining this with $w_5 \leq p_4$ yields $p_4 > p_5$. Hence, in this subcase, the condition $p_4 > p_5$ is automatically implied and need not be imposed separately.

Subcase 1.2: $w_5 \leq 0$.

The new waiting times are

$$w_4^2 = \max(0, w_4) + p_4 + p_5 - \min(0, w_4) - \Delta w_6^-$$

$$
\begin{aligned}
&= p_4 + p_5 - w_5, \\
w_5^2 &= w_5 - \big(p_4 - \min(0, w_4)\big) = w_5 - p_4 + w_4.
\end{aligned}
$$

Since $w_5 \le 0$, we have

$$w_4^2 = p_4 + p_5 - w_5 \ge 0, \qquad w_5^2 = w_5 - p_4 + w_4 \le 0.$$

The change in machine idle time is

$$
\begin{aligned}
\Delta w_{\text{neg}} &= \min(0, w_4^2) + \min(0, w_5^2) - \min(0, w_4) - \min(0, w_5) \\
&= w_5 - p_4.
\end{aligned}
$$

Since $w_5 \le 0 \le p_4$, it follows that $\Delta w_{\text{neg}} \le 0$. Hence, if $w_5 < p_4$, the waiting times of the subsequent jobs increase; if $w_5 = p_4 = 0$, they remain unchanged.

The change in positive waiting time is

$$
\begin{aligned}
\Delta w_{\text{pos}} &= \max(0, w_4^2) + \max(0, w_5^2) - \max(0, w_4) - \max(0, w_5) \\
&= p_4 + p_5 - w_5.
\end{aligned}
$$

Since $w_5 \le 0$, we obtain $\Delta w_{\text{pos}} \ge 0$. Therefore, in this subcase,

$$\Delta w_{\text{pos}} \ge 0.$$

Case 2: $w_4 > 0$.

In this case, from $\max(0, w_4) + p_4 \ge \max(0, w_5)$ it follows that

$$w_5 \le p_4 + w_4.$$

Subcase 2.1: $w_5 \le 0$.

The new waiting times are

$$
\begin{aligned}
w_4^2 &= \max(0, w_4) + p_4 + p_5 - \min(0, w_4) - \Delta w_6^- \\
&= w_4 + p_4 + p_5 - w_5, \\
w_5^2 &= w_5 - \big(p_4 - \min(0, w_4)\big) = w_5 - p_4.
\end{aligned}
$$

Since $w_5 \le 0$, we have

$$w_4^2 = w_4 + p_4 + p_5 - w_5 \ge 0, \qquad w_5^2 = w_5 - p_4 \le 0.$$

The change in machine idle time is

$$
\begin{aligned}
\Delta w_{\text{neg}} &= \min(0, w_4^2) + \min(0, w_5^2) - \min(0, w_4) - \min(0, w_5) \\
&= -p_4.
\end{aligned}
$$

Since $p_4 \ge 0$, we have $\Delta w_{\text{neg}} = -p_4 \le 0$. Hence, if $p_4 > 0$, the waiting times of the subsequent jobs increase; if $p_4 = 0$, they remain unchanged.

The change in positive waiting time is

$$
\begin{aligned}
\Delta w_{\text{pos}} &= \max(0, w_4^2) + \max(0, w_5^2) - \max(0, w_4) - \max(0, w_5) \\
&= p_4 + p_5 - w_5.
\end{aligned}
$$

Since $w_5 \le 0$, it follows that $\Delta w_{\text{pos}} \ge 0$. Therefore, in this subcase,

$$\Delta w_{\text{pos}} \ge 0.$$

Subcase 2.2: $0 < w_5 \le p_4$.

The new waiting times are

$$w_4^2 = \max(0, w_4) + p_4 + p_5 - \min(0, w_4) - \Delta w_6^-$$

$$
\begin{aligned}
&= w_4 + p_4 + p_5 - w_5, \\
w_5^2 &= w_5 - \big(p_4 - \min(0, w_4)\big) = w_5 - p_4.
\end{aligned}
$$

Since $0 < w_5 \le p_4$, we have

$$
w_4^2 = w_4 + p_4 + p_5 - w_5 \ge 0, \qquad w_5^2 = w_5 - p_4 \le 0.
$$

The change in machine idle time is

$$
\begin{aligned}
\Delta w_{\text{neg}} &= \min(0, w_4^2) + \min(0, w_5^2) - \min(0, w_4) - \min(0, w_5) \\
&= w_5 - p_4.
\end{aligned}
$$

Because $w_5 \le p_4$, it follows that $\Delta w_{\text{neg}} \le 0$. Hence, if $w_5 < p_4$, the waiting times of the subsequent jobs increase; if $w_5 = p_4$, they remain unchanged.

The change in positive waiting time is

$$
\begin{aligned}
\Delta w_{\text{pos}} &= \max(0, w_4^2) + \max(0, w_5^2) - \max(0, w_4) - \max(0, w_5) \\
&= p_4 + p_5 + w_4 - w_5 - w_4 - w_5 \\
&= p_4 + p_5 - 2w_5.
\end{aligned}
$$

Therefore,

$$
\Delta w_{\text{pos}} < 0 \iff p_4 + p_5 - 2w_5 < 0 \iff w_5 > \frac{1}{2}(p_4 + p_5).
$$

Thus, under $0 < w_5 \le p_4$, the condition $\Delta w_{\text{pos}} < 0$ is again equivalent to

$$
w_5 > \frac{1}{2}(p_4 + p_5).
$$

As above, combining this inequality with $w_5 \le p_4$ implies $p_4 > p_5$, so the latter condition is automatically satisfied in this subcase.

Subcase 2.3: $p_4 < w_5 \le p_4 + w_4$.

The new waiting times are

$$
\begin{aligned}
w_4^2 &= \max(0, w_4) + p_4 + p_5 - \min(0, w_4) - \Delta w_6^- \\
&= w_4 + p_4 + p_5 - p_4 \\
&= w_4 + p_5, \\
w_5^2 &= w_5 - \big(p_4 - \min(0, w_4)\big) = w_5 - p_4.
\end{aligned}
$$

Since $p_4 < w_5$ and $w_4 > 0$, we have

$$
w_4^2 = w_4 + p_5 \ge 0, \qquad w_5^2 = w_5 - p_4 \ge 0.
$$

The change in machine idle time is

$$
\begin{aligned}
\Delta w_{\text{neg}} &= \min(0, w_4^2) + \min(0, w_5^2) - \min(0, w_4) - \min(0, w_5) \\
&= 0.
\end{aligned}
$$

Hence, the waiting times of the subsequent jobs remain unchanged.

The change in positive waiting time is

$$
\begin{aligned}
\Delta w_{\text{pos}} &= \max(0, w_4^2) + \max(0, w_5^2) - \max(0, w_4) - \max(0, w_5) \\
&= (w_4 + p_5) + (w_5 - p_4) - w_4 - w_5 \\
&= p_5 - p_4.
\end{aligned}
$$

Therefore,

$$
\Delta w_{\text{pos}} < 0 \iff p_5 - p_4 < 0 \iff p_4 > p_5.
$$

Thus, in this subcase, $p_4 > p_5$ is the necessary and sufficient condition for $\Delta w_{\text{pos}} < 0$.

Moreover, under $p_4 < w_5 \le p_4 + w_4$, if $p_4 > p_5$, then automatically

$$w_5 > p_4 > \frac{1}{2}(p_4 + p_5),$$

so the condition $w_5 > \frac{1}{2}(p_4+p_5)$ is automatically satisfied and need not be imposed separately.

Combining the above cases, we see that swapping jobs 4 and 5 can only leave the waiting times of the subsequent jobs unchanged or increase them. Therefore, in order for the swap to improve the total waiting time, it is necessary that

$$\Delta w_{\mathrm{pos}} < 0.$$

Collecting all cases in which $\Delta w_{\mathrm{pos}} < 0$, we obtain

$$\Delta w_{\mathrm{pos}} < 0 \iff \left(w_5 > \frac{1}{2}(p_4 + p_5)\right) \wedge (p_4 > p_5)\,.$$

That is, when jobs are arranged in nondecreasing order of release times, the necessary and sufficient conditions for the adjacent swap to reduce the total waiting time are

$$w_5 > \frac{1}{2}(p_4 + p_5) \qquad \text{and} \qquad p_4 > p_5.$$

Finally, under the condition $\Delta w_{\mathrm{pos}} < 0$, we examine when $\Delta w_{\mathrm{neg}} = 0$. There are three possibilities:

1. If $w_4 \le 0$ and $w_5 = p_4$, then the waiting times of the subsequent jobs remain unchanged, and
$$w_5^2 = w_5 - p_4 + \min(0, w_4) = w_4.$$
2. If $w_4 > 0$ and $w_5 = p_4$, then the waiting times of the subsequent jobs remain unchanged, and
$$w_5^2 = w_5 - p_4 = 0.$$
3. If $w_4 > 0$ and $w_5 > p_4$, then the waiting times of the subsequent jobs remain unchanged, and
$$w_5^2 = w_5 - p_4 > 0.$$

Hence, the above three cases can be unified as

$$w_5^2 \ge \min(0, w_4).$$

□

C.2 Lemma 15 and its proof

Lemma 1 (Joint elimination of over-consumption and under-consumption local optima) *Let the current schedule σ^t be a backward-type local optimum, and suppose that it contains an over-consumption local optimum whose corresponding excessively forward-moved job is $\sigma^t(j)$. Let*

$$B = \{\sigma^t(h_1), \ldots, \sigma^t(h_m)\}$$

denote a set of jobs whose combined forward adjustment replaces the individual forward move of $\sigma^t(j)$. Then, after Algorithm 1 first exposes the relevant local structure explicitly, and Algorithm 3 subsequently moves $\sigma^t(j)$ backward to a position where the discontinuity induced by it can be repaired, at most the following two cases may occur:

1. *no under-consumption local optimum is further induced, in which case the over-consumption local optimum can be eliminated through finitely many backward candidate traversals combined with the adjacent-swap rule;*

2. *an under-consumption local optimum is further induced, in which case the newly exposed local optimum can still be eliminated within the combined neighborhood enumerated by Algorithm 1 and Algorithm 3.*

Therefore, regardless of which case occurs, all subsequent local structures generated by this over-consumption local optimum can ultimately be eliminated successively within the combined neighborhood, thereby further reducing the side effects caused by the increase flow.

Proof Let $\sigma^t(j)$ be the excessively forward-moved job corresponding to the over-consumption local optimum. The essence of this structure is that, under the greedy rule, although the individual forward move of $\sigma^t(j)$ is accepted first at the local level, this move introduces too much machine idle time in the increase-flow region and is therefore not globally optimal. Correspondingly, there exists a set of jobs

$$B = \{\sigma^t(h_1), \ldots, \sigma^t(h_m)\},$$

whose combined forward adjustment can replace the individual forward move of $\sigma^t(j)$, induce less total machine idle time, and hence yield a smaller additional loss caused by the increase flow, resulting in a better overall change in total waiting time.

To eliminate this over-consumption local optimum, Algorithm 1 first makes the relevant local structure explicit. Then Algorithm 3 moves $\sigma^t(j)$ backward, through backward candidate traversal, to a position where the discontinuity induced by it can be reabsorbed. We then distinguish two cases for the increase-flow region induced by this backward move.

Case 1. No under-consumption local optimum is further exposed in the increase-flow region.

In this case, the backward repair of $\sigma^t(j)$ has already removed the main structural imbalance originally caused by its excessive forward move, and the remaining effect in the increase-flow region consists only of additional losses that can be further compressed by local forward adjustments. One may then further realize local forward adjustments of some jobs in the set B through the adjacent-swap rule, thereby replacing the original single-job structure by a combined compensation structure. Since the total machine idle time induced by this combined structure is strictly smaller than that induced by the individual forward move of $\sigma^t(j)$, the total waiting time is strictly reduced. Hence, this over-consumption local optimum is directly eliminated within the combined neighborhood.

Case 2. An under-consumption local optimum is further induced in the increase-flow region.

In this case, there exist some jobs, two representative ones being denoted by $\sigma^t(h_a)$ and $\sigma^t(h_b)$, such that under the current local structure a single adjacent swap is insufficient to strictly reduce the total waiting time corresponding to the increase flow, whereas a better improvement is achieved by moving $\sigma^t(h_a)$ forward across several positions beyond $\sigma^t(h_b)$. In other words, after Algorithm 3 moves $\sigma^t(j)$ backward to a suitable position, a new under-consumption local optimum becomes explicitly exposed in the increase-flow region induced by this move.

The essence of an under-consumption local optimum is precisely that an ideal forward direction admitting strict improvement has not yet been fully released. To handle this structure, Algorithm 1 again first applies a forward move together with the bottleneck breakthrough rule to make the corresponding discontinuity explicit, and Algorithm 3 then further coordinates the local state through backward candidate traversal and adjacent swaps.

Moreover, since the decrease-flow regions corresponding to $\sigma^t(j)$ and $\sigma^t(h_a)$ overlap, Lemma 1 implies that when the decrease flows induced by multiple job moves act on the same region, they must compete with one another because the room for further compressing positive waiting times in that region is limited. Therefore, once $\sigma^t(j)$ and $\sigma^t(h_a)$ cannot coexist, they cannot simultaneously serve as the final dominant improvement structure in the overlapping region; one of them must be replaced by the more effective one. Hence, this newly exposed under-consumption local optimum can likewise be further eliminated within the combined neighborhood.

Combining the above arguments, when handling an over-consumption local optimum, Algorithm 1 first explicitly exposes the relevant structure, and Algorithm 3 then performs the backward repair. If no under-consumption local optimum is further induced, the structure can be eliminated directly through combined forward adjustments; if an under-consumption local optimum is further induced, the newly exposed structure can still be eliminated successively by following the same order, namely, Algorithm 1 first exposes the structure and Algorithm 3 then repairs it. Therefore, regardless of which case occurs, all subsequent local structures generated by the over-consumption local optimum can be successively eliminated within the combined neighborhood, thereby minimizing the side effects caused by the increase flow as much as possible. $\square$