This page is intentionally blank. Proper e-companion title page, with INFORMS branding and exact metadata of the main paper, will be produced by the INFORMS office when the issue is being assembled.

E-Companion

Appendix A: Stochastic Reward Processes

Recall that in the main paper, we assume the reward process is deterministic, with the reward rate $h(t)$ being either time-invariant (core model) or time-varying (extension in Section 7). In this section, we demonstrate that a stochastic reward process can be reformulated as an equivalent deterministic reward process, provided it admits a deterministic compensator. This extension is particularly relevant in applications such as e-commerce order consolidation, where the multi-order cost-saving process for each customer is a stochastic jump process, accruing discrete rewards only when subsequent orders for an existing customer's order arrive.

Specifically, suppose that customer i's accumulated reward process is governed by a jump process $X_i(t)$, which is assumed to be càdlàg (right-continuous with left limits). Let $\mathbb{F}_i = \{\mathcal{F}_{i,t}\}_{t\geq 0}$ denote the natural filtration generated by $X_i(t)$, augmented to satisfy the usual conditions of completeness and right-continuity. To capture the strict non-anticipative online nature of the system, we define the standard global filtration $\mathbb{F} = \{\mathcal{F}_t\}_{t\geq 0}$, where $\mathcal{F}_t \triangleq \sigma\left(\bigcup_{\{i:t_i\leq t\}} \mathcal{F}_{i,t}\right)$ is the σ-algebra generated exclusively by the filtrations of customers who have arrived by time t.

Let $\lambda_i(t)$ denote the arrival intensity of the multi-order process at time t. To ensure a deterministic compensator, we assume $\lambda_i(t)$ is a deterministic càdlàg function, where $\lambda_i(t) = 0$ for all $t < t_i$. Let $w_i(t)$ represent the associated savings per multi-order arrival, drawn i.i.d. from a random variable W_i. The process $X_i(t)$ represents the accumulated reward for customer i up to time t, with $X_i(t) = 0$ for $t < t_i$. We define the deterministic reward rate as $h_i(t) \triangleq \lambda_i(t)\mathbb{E}[W_i]$. Consequently, the integral $\int_0^t h_i(s)ds$ serves as the deterministic compensator for $X_i(t)$, capturing the expected savings accumulated up to time t.

It follows that the compensated process

$$M_i(t) \triangleq X_i(t) - \int_0^t h_i(s)ds$$

is a zero-mean càdlàg martingale with respect to $\mathbb{F}_i$. Because the customers' arrival and reward processes are independent, $M_i(t)$ is also a zero-mean martingale with respect to the standard global filtration $\mathbb{F}$.

The dispatch time $\tau_i \geq t_i$ for order i under any standard non-anticipative online policy is thus a valid stopping time with respect to $\mathbb{F}$. Furthermore, because any order i must be dispatched before its strict deadline (lifespan), the stopping time τ_i is almost surely bounded. By applying Doob's Optional Stopping Theorem to the martingale $M_i(t)$ and the bounded stopping time τ_i, we obtain $\mathbb{E}[M_i(\tau_i)] = \mathbb{E}[M_i(0)] = 0$, which yields

$$\mathbb{E}[X_i(\tau_i)] = \mathbb{E}\left[\int_0^{\tau_i} h_i(s)ds\right] = \mathbb{E}\left[\int_{t_i}^{\tau_i} h_i(s)ds\right].$$

Here, $\mathbb{E}[X_i(\tau_i)]$ represents the expected cost savings physically accumulated through the stochastic multi-order consolidation, while $\mathbb{E}\left[\int_{t_i}^{\tau_i} h_i(s)ds\right]$ denotes the expected value of the accumulated deterministic reward rate $h_i(t)$ up to the dispatch time τ_i. Under this martingale property, the expected actual reward of any online algorithm (including ours) operating under $\mathbb{F}$ is perfectly equivalent to the reward accrued by operating on the deterministic reward-rate functions $h_i(t)$.

For problems governed by a stochastic reward process, we benchmark our policies against a clairvoyant optimal online algorithm. In contrast to the standard online policy, this clairvoyant benchmark operates

under an enlarged global filtration $\mathbb{F}^{\text{clair}} = \{\mathcal{F}_t^{\text{clair}}\}_{t\geq 0}$, formally defined as $\mathcal{F}_t^{\text{clair}} \triangleq \sigma\left(\bigcup_i \mathcal{F}_{i,t}\right)$. By taking the union over all customers i across the entire time horizon, this filtration explicitly provides the exact offline knowledge of all customer arrivals (timing and lifespans) in advance, while the exact multi-order jump realizations $\mathcal{F}_{i,t}$ are still only observed sequentially in time (see, e.g., Golrezaei et al. 2014). Because the individual jump processes are independent of the arrival sequence, $M_i(t)$ remains a valid martingale under $\mathbb{F}^{\text{clair}}$, and the clairvoyant dispatch decisions remain strictly defined as stopping times under $\mathbb{F}^{\text{clair}}$. Consequently, the clairvoyant's expected reward cleanly reduces to that of the optimal offline benchmark (OPT) in our deterministic model, which knows all customers i in the entire time horizon offline, as introduced in Subsection 3.1.

Finally, we note that this abstraction captures a broad class of stochastic behaviors, most notably compound non-homogeneous Poisson processes. In such settings, an individual customer's multi-order arrivals follow a non-homogeneous Poisson process, and the benefit from each realization is an i.i.d. random variable. In its simplest form, this reduces to a standard Poisson process with a constant arrival rate and a fixed jump size representing the cost saved per consolidated order. In conclusion, any stochastic reward process in our setting that admits a deterministic compensator can be rigorously reformulated as an equivalent deterministic reward process with rates $h_i(t)$, justifying the parsimonious reward rate assumptions utilized in our core model.

Appendix B: Extension: Heterogeneous Resources and Soft Capacity Constraints

In this section, we generalize our core framework to accommodate heterogeneous resources and "soft" capacity constraints. This extension captures environments where the system does not face a hard cliff of identical resources, but instead manages a hierarchy of processing tiers (e.g., varying GPU tiers for adaptive LLM inference, or varying virtual machines in cloud computing spot market).

B.1. Generalized Model and Assumptions

Suppose the system manages J heterogeneous resources, indexed by $j \in \{1, \ldots, J\}$. The reward rate for serving a request is now pair-dependent. Specifically, if request i (with intrinsic height h_i) is assigned to resource j, it generates value at a rate of $h_{i,j} = h_i \cdot q_j$, where q_j represents the resource-specific quality. This decomposable (or multiplicative) reward structure has been studied in online allocation and scheduling literature (see, e.g., Charikar et al. (2014)). Without loss of generality, we assume the resources are ordered such that $q_1 \geq q_2 \geq \cdots \geq q_J \geq 0$.

Assumption EC.1 **(Downgrade-Only)**. *Both the online algorithm and the optimal offline benchmark (*OPT*) can only downgrade the resource a request is connected to over time. That is, for any active request i, the index j of the resource serving it must be weakly increasing over time until the request completes or becomes fully unmatched.*

We emphasize that downgrade-only is a restriction coming from the operational environment: it constrains the online algorithm and the optimal offline benchmark identically, so our guarantee is measured against the strongest offline policy that also obeys it. It is, moreover, the direct generalization of the non-resumable preemption of our core model (see the special case below), and it reflects settings in which a request, once moved to a lower-quality resource, is not later upgraded, for example, when re-provisioning or resumption is impractical.

Original Model as a Special Case. Our original model, featuring a hard capacity of C homogeneous resources, is a special case of this generalized framework. By setting $q_j = 1$ for all $j \in \{1, \dots, C\}$, and $q_j = 0$ for all $j > C$, the downgrade-only assumption strictly reduces to the non-resumable preemption assumption. Specifically, once a request's resource index j is downgraded past C, its reward rate permanently drops to 0, which is mathematically equivalent to an irreversible preemption and disconnection from the capacity pool.

Building on this structure, we define $U_\pi(\mathcal{I})$ (and similarly $\mathsf{OPT}(\mathcal{I})$) to be the actual utility achieved under instance $\mathcal{I}$ by policy π. More formally, for any request i, let $j_i^\pi(t)$ denote the index of the resource serving request i at time t under policy π. By Assumption 1, $j_i^\pi(t)$ is weakly increasing over time, implying that the quality tier $q_{j_i^\pi(t)}$ experienced by request i is weakly decreasing. Thus, we formulate the total actual utility as:

$$U_\pi(\mathcal{I}) \triangleq \sum_{i \in [N]} \left(\int_{t_i}^{d_i} q_{j_i^\pi(t)} h_i \, dt \right),$$

where d_i is the time at which request i becomes unmatched under policy π.

B.2. Generalized Index-Based Algorithm

To adapt our index-based policies to this heterogeneous setting, we introduce a greedy cascading assignment mechanism. Let $f_i(t)$ denote the index value of request i at time t as defined in Subsection 3.2. If a resource j is unmatched at time t, we conventionally treat it as serving an active request with an index value of 0.

Algorithm 3 formally characterizes our generalized index-based policy, which triggers a cascading adjustment of assignments whenever a new request is admitted.

Algorithm 3 Generalized Index-Based Policy: Cascading Downgrade Assignment

At any time t when a new request i arrives:

1. **Initial Assignment:** Let S_i be the set of all resources currently serving a request with an index value strictly less than $f_i(t)$. If $S_i = \emptyset$, request i is rejected (left unmatched). Otherwise, allocate request i to the *best* (lowest indexed) server $j_1 \in S_i$.

2. **Cascading Downgrades:** Let i_1 be the request that was displaced from j_1. To downgrade i_1, define S_1 as the set of resources *worse* than j_1 (i.e., subscript $> j_1$) currently serving a request with an index strictly less than $f_{i_1}(t)$.

3. If $S_1 \neq \emptyset$, downgrade i_1 to the best server $j_2 \in S_1$. This displaces a new request i_2.

4. **Iteration:** Repeat this logic for i_2, finding the best available worse server $j_3 \in S_2$, and so forth.

This cascading process continues iteratively until some displaced request i_k finds its corresponding set S_k to be empty. At this point, i_k is left unmatched, and the assignment adjustment process at time t concludes.

B.3. Theoretical Analysis

To establish the competitive ratio of Algorithm 3, we leverage the parallel partitioning framework introduced in Section 4. Specifically, consider a parallel partitioning $\mathcal{B}$ of the plane, with r_{z_i} denoting the unique base curve that passes through the point (t_i, h_i), where t_i is the arrival time of request i and h_i is its intrinsic value. Because we are currently focusing on the case where the intrinsic value h_i remains constant over time, we impose the mild restriction that our base curves must be weakly monotonically decreasing over time.

Let γ_a and γ_o be defined exactly as in Section 4, based *solely* on the intrinsic reward of the requests (i.e., factoring out the multiplicative quality q_j of the actual utility):

$$\begin{aligned}
\gamma_o &\triangleq \inf_{z=(t,h)\in\mathbb{R}^2_+,\ t'\in(t,t+1)} \left\{ \frac{\int_t^{t'} \mathcal{B}_{r_z}(t'')\ dt''}{h(t'-t)} \right\} = \inf_{z=(t,h)\in\mathbb{R}^2_+} \left\{ \frac{\int_t^{t+1} \mathcal{B}_{r_z}(t'')\ dt''}{h} \right\}, \\
\gamma_a &\triangleq \inf_{z=(t,h)\in\mathbb{R}^2_+} \left\{ \inf_{t'\in(t,t+1)} \left\{ \frac{h(t'-t)}{\int_t^{t'} \mathcal{B}_{r_z}(t'')\ dt''} \right\},\ \frac{h}{\int_t^{t+2} \mathcal{B}_{r_z}(t'')\ dt''} \right\} \\
&= \inf_{z=(t,h)\in\mathbb{R}^2_+} \left\{ 1,\ \frac{h}{\int_t^{t+2} \mathcal{B}_{r_z}(t'')\ dt''} \right\},
\end{aligned}$$

where both equalities naturally follow because $\mathcal{B}_{r_z}(t'')$ is weakly monotonically decreasing with respect to t''; thus, the infimum in the top expression is attained as $t' \to t+1$, while in the bottom expression, it occurs as $t' \to t$.

Let $\pi^{\mathcal{B}}$ denote the generalized index-based policy (Algorithm 3) equipped with the index function $f_i(t) = \mathcal{B}_{r_{z_i}}(t)$ for all requests i at any time $t \in \mathbb{R}$. Analogous to Theorem 1, our objective is to prove that $U_{\pi^{\mathcal{B}}}(\mathcal{I}) \geq \gamma_a\gamma_o \mathsf{OPT}(\mathcal{I})$ for all $\mathcal{I}$. To accomplish this, we must first adapt the concepts of quasi-utility and the extended policy to our generalized heterogeneous setting.

First, akin to Definition 6, the quasi-utility of any policy π under any instance $\mathcal{I}$ is defined as:

$$V_{\pi,\mathcal{B}}(\mathcal{I}) \triangleq \sum_{i\in[N]} \left(\int_{t_i}^{d_i} q_{j_i^\pi(t)} \mathcal{B}_{r_{z_i}}(t)\ dt \right).$$

Second, the extended policy $\tilde{\pi}$ is defined analogously to Definition 5, with one key distinction: we extend any completed request (i.e., conceptually continue serving it for one extra unit of time) on the *final* resource it occupied prior to completion. Note that this final resource inherently possesses the lowest quality among all resources that the request was ever assigned to.

Paralleling Subsection 4.2, the proof of Theorem 1 in this generalized setting relies on three key lemmas.

LEMMA EC.1. *For any instance $\mathcal{I}$, any downgrade-only policy π, and (weakly decreasing) base $\mathcal{B}$, we have*

$$U_\pi(\mathcal{I}) \geq \gamma_a V_{\tilde{\pi},\mathcal{B}}(\mathcal{I}).$$

Proof. The proof closely mirrors its homogeneous counterpart in Subsection D.1.1. In particular, it is sufficient to show that for all $i \in [N]$, we have

$$\int_{t_i}^{d_i} q_{j_i^\pi(t)} h_i\ dt \geq \gamma_a \left(\int_{t_i}^{\tilde{d}_i} q_{j_i^\pi(t)} \mathcal{B}_{r_{z_i}}(t)\ dt \right).$$

We partition the analysis into two cases:

1. $d_i < t_i + 1$: In this case, $\tilde{d}_i = d_i$, and we can write
$$\frac{\int_{t_i}^{d_i} q_{j_i^\pi(t)} h_i \ dt}{\int_{t_i}^{d_i} q_{j_i^\pi(t)} \mathcal{B}_{r_{z_i}}(t) \ dt} \geq 1 \geq \gamma_a.$$
This holds because the ratio $\frac{q_{j_i^\pi(t)} h_i}{q_{j_i^\pi(t)} \mathcal{B}_{r_{z_i}}(t)} = \frac{h_i}{\mathcal{B}_{r_{z_i}}(t)}$ is weakly increasing as a function of t, meaning the infimum is strictly bounded from below as $d_i \to t_i$.
2. $d_i = t_i + 1$: In this case, $\tilde{d}_i = d_i + 1 = t_i + 2$, yielding:
$$\begin{aligned}
\frac{\int_{t_i}^{t_i+1} q_{j_i^\pi(t)} h_i \ dt}{\int_{t_i}^{t_i+2} q_{j_i^\pi(t)} \mathcal{B}_{r_{z_i}}(t) \ dt} &= \frac{\int_{t_i}^{t_i+1} (q_{j_i^\pi(t)} - q_{j_i^\pi(t_i+1)}) h_i \ dt \ + \ q_{j_i^\pi(t_i+1)} h_i}{\int_{t_i}^{t_i+2} (q_{j_i^\pi(t)} - q_{j_i^\pi(t_i+1)}) \mathcal{B}_{r_{z_i}}(t) \ dt \ + \ q_{j_i^\pi(t_i+1)} \int_{t_i}^{t_i+2} \mathcal{B}_{r_{z_i}}(t) \ dt} \\
&= \frac{\int_{t_i}^{t_i+1} (q_{j_i^\pi(t)} - q_{j_i^\pi(t_i+1)}) h_i \ dt \ + \ q_{j_i^\pi(t_i+1)} h_i}{\int_{t_i}^{t_i+1} (q_{j_i^\pi(t)} - q_{j_i^\pi(t_i+1)}) \mathcal{B}_{r_{z_i}}(t) \ dt \ + \ q_{j_i^\pi(t_i+1)} \int_{t_i}^{t_i+2} \mathcal{B}_{r_{z_i}}(t) \ dt} \\
&\geq \min \left\{ \frac{\int_{t_i}^{t_i+1} (q_{j_i^\pi(t)} - q_{j_i^\pi(t_i+1)}) h_i \ dt}{\int_{t_i}^{t_i+1} (q_{j_i^\pi(t)} - q_{j_i^\pi(t_i+1)}) \mathcal{B}_{r_{z_i}}(t) \ dt}, \frac{q_{j_i^\pi(t_i+1)} h_i}{q_{j_i^\pi(t_i+1)} \int_{t_i}^{t_i+2} \mathcal{B}_{r_{z_i}}(t) \ dt} \right\} \\
&\geq \min \left\{ 1, \frac{h_i}{\int_{t_i}^{t_i+2} \mathcal{B}_{r_{z_i}}(t) \ dt} \right\} \geq \gamma_a.
\end{aligned}$$
Here, the second equality utilizes the fact that the request is conceptually extended on its final resource for the entire duration from $t_i + 1$ to $t_i + 2$. The third inequality leverages the algebraic property that the ratio of sums is bounded by the minimum and maximum of the corresponding individual ratios. The final inequality holds because all coefficients of $(q_{j_i^\pi(t)} - q_{j_i^\pi(t_i+1)})$ are non-negative due to downgrade-only assumption, and the ratio $\frac{h_i}{\mathcal{B}_{r_{z_i}}(t)}$ is weakly increasing as a function of t.

This concludes the proof. □

LEMMA EC.2. *For any instance $\mathcal{I}$ and base $\mathcal{B}$, we have*
$$V_{\widetilde{\pi^{\mathcal{B}}},\mathcal{B}}(\mathcal{I}) \geq V_{\mathsf{OPT},\mathcal{B}}(\mathcal{I}).$$

Proof. The proof of this lemma expands upon its homogeneous counterpart (Lemma 3), albeit with additional structural intricacies. For notational simplicity in this proof, we refer to $\widetilde{\pi^{\mathcal{B}}}$ simply as $\tilde{\pi}$. We evaluate the instantaneous quasi-utility rate of both algorithms at an arbitrary time t. Specifically, we show that the summation of the base curve values (weighted by their matched resource qualities) for the requests served by $\tilde{\pi}$ is at least that of the requests served by OPT at any given time t:
$$\sum_{i \in \mathcal{H}_{\tilde{\pi}}(t)} q_{j_i^{\tilde{\pi}}(t)} \mathcal{B}_{r_i}(t) \geq \sum_{i \in \mathcal{H}_{\mathsf{OPT}}(t)} q_{j_i^{\mathsf{OPT}}(t)} \mathcal{B}_{r_i}(t). \tag{EC.1}$$
Because OPT can serve at most one request per resource, let $i_j^{\mathsf{OPT}}(t) \in \mathcal{H}_{\mathsf{OPT}}(t)$ denote the unique request that OPT is serving on resource j at time t. Similarly, let $i_j^{\tilde{\pi}}(t) \triangleq \{i \in \mathcal{H}_{\tilde{\pi}}(t) | j_i^{\tilde{\pi}}(t) = j\}$ denote the set of all requests that policy $\tilde{\pi}$ is associating with resource j at time t. Note that $i_j^{\tilde{\pi}}(t)$ may encompass a single actual active request, alongside potentially multiple extended requests that completed at resource j during the interval $(t-1, t]$. Using this notation, we can rewrite Equation (EC.1) as:
$$\sum_{j \in [J]} \left(q_j \sum_{i \in i_j^{\tilde{\pi}}(t)} \mathcal{B}_{r_i}(t) \right) \geq \sum_{j \in [J]} q_j \mathcal{B}_{r_{i_j^{\mathsf{OPT}}(t)}}(t). \tag{EC.2}$$
Suppose we sort the base indices $\{r_i\}_{i \in \mathcal{H}_{\mathsf{OPT}}(t)}$ in descending order, denoted by the sequence $(r_{\sigma_1}, r_{\sigma_2}, \ldots, r_{\sigma_J})$. We assert the following claim:

CLAIM EC.1. *At any time t and $\forall j \in [J]$, we have*

$$\max_{i \in i_j^{\tilde{\pi}}(t)} r_i \geq r_{\sigma_j},$$

which immediately implies that $\max_{i \in i_j^{\tilde{\pi}}(t)} \mathcal{B}_{r_i}(t) \geq \mathcal{B}_{r_{\sigma_j}}(t)$.

Assuming this claim holds, we can readily establish Equation (EC.2) through the following chain of inequalities:

$$\sum_{j\in[J]} \left(q_j \sum_{i \in i_j^{\tilde{\pi}}(t)} \mathcal{B}_{r_i}(t) \right) \geq \sum_{j\in[J]} \left(q_j \max_{i \in i_j^{\tilde{\pi}}(t)} \mathcal{B}_{r_i}(t) \right) \geq \sum_{j\in[J]} q_j \mathcal{B}_{r_{\sigma_j}}(t) \geq \sum_{j\in[J]} q_j \mathcal{B}_{r_{i_j^{\mathsf{OPT}}(t)}}(t).$$

Here, the first inequality relies on the non-negativity of the base curves; the second directly applies the claim; and the final inequality leverages the established rearrangement inequality—sorting two non-negative sequences in descending order uniquely maximizes their pair-wise dot product across all possible permutations.

We now prove the central claim.

Proof of Claim EC.1. We proceed by contradiction. Suppose there exists a resource $j \in [J]$ at time t such that:

$$\max_{i \in i_j^{\tilde{\pi}}(t)} r_i < r_{\sigma_j}.$$

Observe two structural properties maintained by our generalized index-based policy (Algorithm 3): (1) *Weak Monotonicity*: the index of the base curve assigned to any resource j is weakly increasing over time (prior to completion), because replacements exclusively occur via higher-indexed requests; and (2) *Greedy Cascading*: any displaced request is systematically pushed to the best available lower-tier resource among those serving a weakly lower base value.

By the Weak Monotonicity property, if at any time $t' \in (t-1, t]$ our policy is serving a request i with base index r_i on resource j, then either this request is successively replaced by requests with weakly higher base indices, or it completes at some time $t'' \in [t', t]$. In the event of completion, the extended policy $\tilde{\pi}$ sustains its value for an additional unit of time, extending beyond t. Formally, this yields:

$$\max_{t' \in (t-1,t]} r_{i_j^{\pi^{\mathcal{B}}}(t')} = \max_{i \in i_j^{\tilde{\pi}}(t)} r_i < r_{\sigma_j}.$$

Given this inequality, and knowing that the j requests corresponding to $(r_{\sigma_1} \geq \cdots \geq r_{\sigma_j})$ must have arrived within the interval $(t-1, t]$, it follows that these j requests have possessed strictly higher base indices than the occupying requests at resource j for their entire duration up to time t. Consequently, all j of these requests must have been assigned to resources of a strictly higher tier than j upon arrival. Even if they experienced subsequent downgrades, they must have remained strictly above resource j up to time t; had they ever dropped out of top $j-1$ resources, the Greedy Cascading property would dictate they should first displace the lower-valued request at resource j, as j is always their preferred resource compared to anything other than top $j-1$ resources.

This creates an unavoidable contradiction: there are only $j-1$ resources better than j, yet they must simultaneously accommodate all j requests corresponding to $(r_{\sigma_1}, \ldots, r_{\sigma_j})$. This violates the hard constraint that each resource serves at most one request at a time, completing the proof of the claim. □

LEMMA EC.3. *For any instance $\mathcal{I}$ and (weakly decreasing) base $\mathcal{B}$, we have*

$$V_{\mathsf{OPT},\mathcal{B}}(\mathcal{I}) \geq \gamma_o \mathsf{OPT}(\mathcal{I}).$$

Proof. The logic closely tracks the homogeneous version in Subsection D.1.3. Specifically, it suffices to prove that for all $i \in [N]$, we have:

$$\int_{t_i}^{d_i^*} q_{j_i^{\mathsf{OPT}}(t)} \mathcal{B}_{r_{z_i}}(t)\ dt \geq \gamma_o \left(\int_{t_i}^{d_i^*} q_{j_i^{\mathsf{OPT}}(t)} h_i\ dt \right).$$

Crucially, we note two facts: (i) the ratio $\frac{\mathcal{B}_{r_{z_i}}(t)}{h_i}$ is a weakly decreasing function of t, and (ii) the quality multiplier $q_{j_i^{\mathsf{OPT}}(t)}$ is a weakly decreasing function of t due to the downgrade-only assumption. Therefore, we can write:

$$\frac{\int_{t_i}^{d_i^*} q_{j_i^{\mathsf{OPT}}(t)} \mathcal{B}_{r_{z_i}}(t)\ dt}{\int_{t_i}^{d_i^*} q_{j_i^{\mathsf{OPT}}(t)} h_i\ dt} \geq \frac{\int_{t_i}^{d_i^*} \mathcal{B}_{r_{z_i}}(t)\ dt}{\int_{t_i}^{d_i^*} h_i\ dt} \geq \frac{\int_{t_i}^{t_i+1} \mathcal{B}_{r_{z_i}}(t)\ dt}{\int_{t_i}^{t_i+1} h_i\ dt} \geq \gamma_o,$$

which validates the bound and completes the proof. □

By sequentially chaining the inequalities established in Lemmas EC.1, EC.2, and EC.3, we establish the generalized counterpart to Theorem 1.

THEOREM EC.1. *Under the generalized heterogeneous resource model, for any (weakly decreasing) base $\mathcal{B}$, the generalized index-based policy $\pi^{\mathcal{B}}$ (Algorithm 3) achieves a competitive ratio of at least $\gamma_a\gamma_o$. That is,*

$$U_{\pi^{\mathcal{B}}}(\mathcal{I}) \geq \gamma_a\gamma_o \mathsf{OPT}(\mathcal{I}) \quad \forall \mathcal{I}.$$

Proof. Combining Lemmas EC.1, EC.2, and EC.3 yields the required sequence of bounds:

$$U_{\pi^{\mathcal{B}}}(\mathcal{I}) \geq \gamma_a V_{\widetilde{\pi^{\mathcal{B}}},\mathcal{B}}(\mathcal{I}) \geq \gamma_a V_{\mathsf{OPT},\mathcal{B}}(\mathcal{I}) \geq \gamma_a\gamma_o \mathsf{OPT}(\mathcal{I}). \quad \square$$

This directly guarantees the competitive ratio $\gamma_a\gamma_o$, seamlessly extending our analytical framework to environments with heterogeneous resources and soft capacity. Consequently, we can directly import the exponential base curves from Subsection 4.3 to achieve the identical competitive ratios established in the homogeneous setting. Specifically, let $\pi^{\exp(\alpha^*)}$ denote the generalized index-based policy (Algorithm 3) operating with the exponential index function:

$$f_i(t) \triangleq h_i e^{-\alpha^*(t-t_i)},$$

where α^* is defined as in Equation (10). We then obtain the following performance guarantee:

COROLLARY EC.1. *The competitive ratio $\mathcal{CR}$ of the generalized index-based policy $\pi^{\exp(\alpha^*)}$ satisfies:*

$$\mathcal{CR}(\pi^{\exp(\alpha^*)}) = \frac{1-e^{-\alpha^*}}{\alpha^*} \geq 0.68.$$

REMARK EC.1 (ROBUSTNESS TO DYNAMIC RESOURCE PROFILES). As a direct generalization of Remark 2 from the homogeneous setting, our competitive ratio guarantees for the heterogeneous model remain rigorously robust to dynamic capacity environments. While our exposition assumes a static hierarchy of J resources, our generalized analysis relies exclusively on instantaneous, pointwise bounds on the quasi-utility rate (e.g., Lemma EC.2). Consequently, our results hold even if the availability (capacity) or the quality q_j of the various resource tiers fluctuates arbitrarily over time, provided the algorithm dynamically adjusts assignments according to the current capacity profile while strictly adhering to the downgrade-only restriction (i.e., active requests can only be shifted to weakly lower-tier resources or permanently discharged).

REMARK EC.2 (COMPATIBILITY WITH TIME-VARYING REWARDS). The proof framework developed in this section focuses on constant intrinsic heights h_i strictly for simplicity of exposition. By defining the appropriate exponential base curves analogously to Section 7, this heterogeneous resource extension can be generalized to the time-varying reward rates setting of Section 7.

This extension illustrates that our framework and guarantees are robust to a particular heterogeneous-resource (soft-capacity) structure, in which a request displaced from a higher-quality tier may continue on a lower one. It is distinct from adaptively choosing overflow capacity at an explicit cost or penalty, which we do not claim to resolve and regard as a separate, interesting direction for future work.

Appendix C: Proofs of Section 3

C.1. Proofs in Subsection 3.3

C.1.1. Proof of Lemma 1

Proof of Lemma 1. To prove this, we first provide an adjustment to the algorithm π^{h}, namely $\widetilde{\pi^{\mathsf{h}}}$. The departure decisions in $\widetilde{\pi^{\mathsf{h}}}$ are exactly same as in π^{h}, except for requests that π^{h} keeps until their deadline, i.e., for a whole duration of 1. For such requests, we assume that $\widetilde{\pi^{\mathsf{h}}}$ keeps them twice as much, i.e., for a total duration of 2. Notice that neither the capacity constraint nor the deadline constraint needs to be satisfied for $\widetilde{\pi^{\mathsf{h}}}$, as this policy is only for the sake of analysis and will not be used for implementation. Lastly, we will prove the following two inequalities:

1. $2\, U_{\pi^{\mathsf{h}}}(\mathcal{I}) \geq U_{\widetilde{\pi^{\mathsf{h}}}}(\mathcal{I})$:

The proof of this inequality simply comes from the definition of $U_{\widetilde{\pi^{\mathsf{h}}}}(\mathcal{I})$, as all requests that are preempted before their deadline under π^{h} will be served for the same duration under $\widetilde{\pi^{\mathsf{h}}}$ as well, and all those which are kept until their deadline, i.e., for duration 1, will now be kept for duration 2 under $\widetilde{\pi^{\mathsf{h}}}$. Therefore, for each request, $\widetilde{\pi^{\mathsf{h}}}$ will get at most twice accumulated reward as π^{h}, now if we sum over all requests we will get the inequality we wanted.

2. $U_{\widetilde{\pi^{\mathsf{h}}}}(\mathcal{I}) \geq \mathsf{Skyline}(\mathcal{I})$:

We claim that at any time t, $\widetilde{\pi^{\mathsf{h}}}$ is serving requests with rewards that dominates that of $\mathsf{Skyline}(\mathcal{I})$. More specifically, at any time t, let $h_{s_1} \geq h_{s_2} \geq ... \geq h_{s_k}, (k \leq C)$, be the distinct requests that $\mathsf{Skyline}(\mathcal{I})$ is keeping. Then we claim that there are distinct requests $h_{i_1} \geq h_{i_2} \geq ... \geq h_{i_k}$ that $\widetilde{\pi^{\mathsf{h}}}(\mathcal{I})$ is serving which satisfies $\forall 1 \leq j \leq k,\ h_{i_j} \geq h_{s_j}$.

First notice that if $\pi^{\mathsf{h}}(\mathcal{I})$ is serving some request at time t, $\widetilde{\pi^{\mathsf{h}}}(\mathcal{I})$ does so as well. Now among the sequence $h_{s_1} \geq h_{s_2} \geq ... \geq h_{s_k}$ consider the request with highest reward that $\pi^{\mathsf{h}}(\mathcal{I})$ is not serving at time t, say s_j. This means that $\pi^{\mathsf{h}}(\mathcal{I})$ must have preempted request s_j at some time $t' \leq t$. This implies that, at the time t', $\pi^{\mathsf{h}}(\mathcal{I})$ was keeping C requests all with rewards at least h_{s_j}. Now, look at the time interval $[t', t]$. First, we know that, unless hitting the deadline, $\pi^{\mathsf{h}}(\mathcal{I})$ will only replace a request with another one with higher reward rate. Second, if a request hit its deadline while $\pi^{\mathsf{h}}(\mathcal{I})$ was keeping it, then $\widetilde{\pi^{\mathsf{h}}}(\mathcal{I})$ will keep it for a whole extra unit of time. Therefore, any of such higher reward requests that $\pi^{\mathsf{h}}(\mathcal{I})$ is keeping at time t' will either remain to be served until t, replace with some higher reward (could be multiple replaces over the time interval until t, but each of them has higher reward rate with the previous one in this "replacement chain"), or hit the deadline. Notice that in all of these three cases, as $t - t' < 1$, $\widetilde{\pi^{\mathsf{h}}}(\mathcal{I})$ will serve at least C number of

requests at time t which weakly dominates (in terms of their reward rate) those ones that $\pi^{\mathsf{h}}(\mathcal{I})$ was serving at time t', which themselves weakly dominated the s_j that $\mathsf{Skyline}(\mathcal{I})$ is serving at time t.
Finally, for $s_1, ..., s_{j-1}$ we knew that $\pi^{\mathsf{h}}(\mathcal{I})$ and so as $\widetilde{\pi}^{\mathsf{h}}(\mathcal{I})$ are serving them until at least time t (by definition of s_j). Combining this with the statement that we showed earlier - i.e., $\widetilde{\pi}^{\mathsf{h}}(\mathcal{I})$ is keeping at least C requests with rewards at least h_{s_j} (which is greater or equal than the reward rate of the rest of the requests s_{j+1} to s_k) - we can conclude the proof of this dominance claim (and so as the second inequality) by setting $i_l = s_l,\ \forall 1 \le l \le j-1$, and then choosing $k-(j-1)$ highest reward requests from the rest of the requests that $\widetilde{\pi}^{\mathsf{h}}$ is serving at time t - each of which had a reward rate of at least h_{s_j}.

□

C.1.2. Proof of Proposition 1

Proof of Proposition 1. The proof of $\mathcal{CR}(\pi^{\mathsf{h}}) \ge \frac{1}{2}$ is readily attained from Lemma 1. The rest is to show that this ratio is tight, by providing a simple bad instance. Instance $\mathcal{I}$ consists of only 2 types of requests, where each type has C identical requests. The first type is $z_1 := (t_1 = 0, h_1 = 1)$, and the second type is $z_2 := (t_2 = 1^-, h_2 = 1^-)$. Under this instance $\mathcal{I}$, π^{h} will only serve all the first types till their deadline ($t = 1$) and preempts all second types immediately, whereas the optimal offline $\mathsf{OPT}(\mathcal{I})$ will preempt all the first types when second types arrive and serve them till their deadline ($t = 2^-$). It is clear to see that $U^{\mathsf{h}}_{\pi}(\mathcal{I}) = C$ and $\mathsf{OPT}(\mathcal{I}) = C^- + C^-$, which results in:

$$\mathcal{CR}(\pi^{\mathsf{h}}) \le \frac{U^{\mathsf{h}}_{\pi}(\mathcal{I})}{\mathsf{OPT}(\mathcal{I})} = \frac{1}{2},$$

concluding the tightness of the analysis, i.e., $\mathcal{CR}(\pi^{\mathsf{h}}) = \frac{1}{2}$. □

C.2. Proofs in Subsection 3.4

We will prove Proposition 2 in this subsection.

First, we will provide the definitions and notations that we use to analyse the competitive ratio of the policy π^{ra}. It is true that the algorithm has no non-trivial information about how the optimal offline policy is making its decisions; nonetheless, we can use the decisions of the optimal offline in order to provide fitting structure for our competitive ratio analysis of π^{ra} against the unknown-to-algorithm optimal offline.

In the following we will construct a continuum of bijective functions, $\forall t \in \mathbb{R}$, $\mu_t : \mathcal{H}_{\mathsf{OPT}}(t) \to \mathcal{H}_{\pi^{\mathsf{ra}}}(t)$, which are one-to-one correspondences that match each request that is being served by OPT at time t, with a distinct request that is being served by π^{ra} at the same time t. The main purpose of this one-to-one correspondence is that it provides us with a clear way of accounting the rewards for the requests that π^{ra} is serving and then comparing them with the rewards for the requests that OPT is serving them, without being prone to double counting mistakes. We also denote the inverse of this bijective matching at each time t by $\mu_t^- : \mathcal{H}_{\pi^{\mathsf{ra}}}(t) \to \mathcal{H}_{\mathsf{OPT}}(t)$.

Now let us show how we will construct this one-to-one correspondence at each time. Notice that whenever a new request arrives, as long as there is some capacity available, both of OPT and π^{ra} will start serving that. Additionally, both of these policies face the same instance, meaning that both of them will see the same arrival time and lifespan for each request. This will tell us that since time $t = 0$, as long as the capacity constraint is not binding, both of them will serve the exact same set of requests at each time. Let

us denote t_0 as the first time that the capacity constraint will become binding. With this in mind, we set $\mu_t(i) = i \in \mathcal{H}_{\pi^{\mathsf{ra}}}(t), \forall t \in [0, t_0), i \in \mathcal{H}_{\mathsf{OPT}}(t)$.

In the time intervals that there is no change to either of $\mathcal{H}_{\pi^{\mathsf{ra}}}(t)$ and $\mathcal{H}_{\mathsf{OPT}}(t)$, we also do not change the matching $\mu_t(.)$. Also, notice that whenever a request hits its deadline either of the two policies has to remove it, but we can think of it as they are replacing it with a request with zero reward. Moreover, the index (remaining area) of such a request at the time of its deadline is also zero. Therefore, as for the π^{ra}, we can even view this removal as a deliberate action to a request with same zero index (i.e., to a zero reward request) rather than viewing it such that the algorithm is forced to remove that request whose lifespan is finished. With this in mind, the remaining part for us to show in order to fully characterize $\mu_t(.)$ at all times is to show how $\mu_t(.)$ will change at times when either π^{ra} or OPT are accepting a new arrival.

Suppose request i arrives to the system. There are 4 possible cases that can happen, and we will explicitly state how the matching should adapt in each of them.

1. Neither OPT nor π^{ra} accepts request i: In this case, μ_{t_i} continues to be what it was before t_i without any change.

2. Only OPT accepts i: In this case, consider the request which would be replaced with i by OPT at time t_i. Let us call that request j. The matching at this time, μ_{t_i}, will change in the following manner: we connect i to what j was matched to, i.e., $\mu_{t_i}(i) \leftarrow \mu_{t_i^-}(j)$, and we do not change anything else.

3. Only π^{ra} accepts i: In this case, we do a similar change as in the previous case, but now on the π^{ra} side. More specifically, let j to be the request that is being replaced with i by π^{ra} at time t_i. Also let k to be the match of j right before time t_i, i.e., $k = \mu^-_{t_i^-}(j)$. Then, we will change the match of k at t_i from j to be i, $\mu_{t_i}(k) \leftarrow i$. Again, everything else will remain unchanged.

4. Both π^{ra} and OPT accepts i: In this case, let j_π and j_{OPT} be the requests that are being replaced with i at time t_i by π^{ra} and OPT, respectively. There are two possible scenarios that might have happened:

   (a) $\mu_{t_i^-}(j_{\mathsf{OPT}}) = j_\pi$: In this scenario, both of these requests will be discarded by the corresponding policies, and both policies will accept i. As a result the only change we need to do is to set $\mu_{t_i}(i) = i$.

   (b) $\mu_{t_i^-}(j_{\mathsf{OPT}}) \neq j_\pi$: In this scenario, not only do we need to set $\mu_{t_i}(i) = i$, but also we have to match together the matches of both j_{OPT} and j_π, i.e., $\mu_{t_i}(k) \leftarrow \mu_{t_i^-}(j_{\mathsf{OPT}})$ where $k = \mu^-_{t_i^-}(j_\pi)$.

Notice that we also discard the requests from domain of the matching for requests that get preempted since their departure time.

REMARK EC.3. Note that from the way we construct matching μ_t, we can easily deduce that it would be, indeed, a one-to-one correspondence from $\mathcal{H}_{\mathsf{OPT}}(t)$ to $\mathcal{H}^{\mathsf{ra}}_{\pi}(t)$.

Now that we have built our matching μ_t over time, let us also define another function that determines for any request i what is the sequence of its matched requests (on the π^{ra}'s side) over the time interval it was being served by OPT. More formally, for any request i we define $\sigma_i : [t_i, d_i^*) \to [N]$, where d_i^* was the departure time of i by OPT. For any request i, we will set:

$$\sigma_i(t) \triangleq \mu_t(i), \quad \forall t \in [t_i, d_i^*).$$

With this way of matching the requests that OPT is keeping with the ones that π^{ra} is keeping, we will have the following properties hold:

CLAIM EC.2. *For any request $i \in [N]$ such that $t_i < d_i^*$, the following two properties hold for the requests in the sequence $\sigma_i(.)$:*

1. *At time t_i, the remaining area of the request $\sigma_i(t_i)$ is at least as large as that of request i:*

$$h_{\sigma_i(t_i)}(t_{\sigma_i(t_i)} + 1 - t_i) \geq h_i.$$

2. *At any time $t \in (t_i, d_i^*)$, the remaining area of the request $\sigma_i(t)$ at the time t is at least as large as that of the request $\sigma_i(t^-)$:*

$$h_{\sigma_i(t)}(t_{\sigma_i(t)} + 1 - t) \geq h_{\sigma_i(t^-)}(t_{\sigma_i(t^-)} + 1 - t).$$

Proof of Claim EC.2. The proof follows from the way we have constructed the matching, as well as the fact that the algorithm always keeps the requests with the highest remaining area. Now let us formally state the proof for each part of the claim:

1. Request i arrives at time t_i, now there are two cases:

(a) if algorithm also accepts to serve i (case 4.a): in this case by definition we have $\sigma_i(t_i) = i$ and the statement of this part holds clearly.

(b) if algorithm never serve request i (immediate preempt of i by algorithm, case 2): in this case, we did $\sigma_i(t_i) = \mu_{t_i}(i) \leftarrow \mu_{t_i^-}(j)$. From the fact that request $\mu_{t_i^-}(j)$ is still kept in the system and request i was preempted by π^{ra}, we get that its remaining area at time t_i should have been higher, which is exactly what we wanted to show.

2. To show this part, if $\sigma_i(t^-) = \sigma_i(t)$, then clearly the expression holds with equality. On the other hand, if $\sigma_i(t^-) \neq \sigma_i(t)$, it means that either case 3 or case 4.b) has happened, and in both of them π^{ra} preempts $\sigma_i(t^-)$ instead of $\sigma_i(t)$. Thus, again by the fact that π^{ra} keeps the requests with higher remaining area at t, we immediately get: $h_{\sigma_i(t)}(t_{\sigma_i(t)} + 1 - t) \geq h_{\sigma_i(t^-)}(t_{\sigma_i(t^-)} + 1 - t)$.

□

The next lemma will provide a comparison between the total reward that OPT is acquiring from any request i and the reward that π^{ra} is accumulating over that period of time from the requests that are matched to i, namely $\sigma_i(.)$.

LEMMA EC.4. *For any request $i \in [N]$ such that $t_i < d_i^*$, the following inequality holds:*

$$\int_{t_i}^{d_i^*} h_{\sigma_i(t)}\ dt \geq (1 - \frac{1}{e}) \times h_i(d_i^* - t_i).$$

Proof of Lemma EC.4. To prove this lemma, we make use of Claim EC.2, but before that let us define some simplifying notation. In the sequence of requests $\sigma_i(.)$, let us define τ_j as the j-th changing time in the sequence, with $\tau_0 \triangleq t_i$ and $\tau_{n_i} \triangleq d_i^*$, where n_i is the total number of matches to request i. In other words, τ_j is the j-th time in the sequence that $\sigma_i(\tau_j) \neq \sigma_i(\tau_j^-)$. Let us also define $\forall 1 \leq j \leq n_i : \alpha_j \triangleq \tau_j - \tau_{j-1}, \beta_j \triangleq (1 + t_{\sigma_i(\tau_{j-1})} - \tau_{j-1})$, where α_j would be the length of time that i is matched to its j-th match and β_j would be the remaining lifespan of i's j-th match at the time when its matched interval starts (at time τ_{j-1}). Additionally, for conciseness we refer to $h_{\sigma_i(\tau_{j-1})}$ as h_{i_j}. With these in mind, from Claim EC.2 we have:

$$h_{i_1}\beta_1 \geq h_i,$$
$$h_{i_{j+1}}\beta_{j+1} \geq h_{i_j}(\beta_j - \alpha_j) = h_{i_j}\beta_j(1 - \frac{\alpha_j}{\beta_j}) \quad \forall 1 \leq j \leq n_i - 1,$$

where the first inequality is equivalent to the first part of the claim, and the second inequality will be derived by setting $t=\tau_j$ for all $1\le j\le n_i-1$, in the second part of the claim. By using the second inequality in a cascade manner, we can get:

$$\forall 1\le j\le n_i-1:\ h_{i_{j+1}}\beta_{j+1}\ge h_{i_1}\beta_1\prod_{k=1}^{j}(1-\frac{\alpha_k}{\beta_k})\ \Rightarrow\ h_{i_{j+1}}\alpha_{j+1}\ge h_{i_1}\beta_1\left(\prod_{k=1}^{j}(1-\frac{\alpha_k}{\beta_k})\right)\frac{\alpha_{j+1}}{\beta_{j+1}}.$$

Then, if we sum up the above inequality for all j from 0 to n_i-1 we get:

$$\begin{aligned}\int_{t_i}^{d_i^*}h_{\sigma_i(t)}\ dt&=\sum_{j=0}^{n_i-1}h_{i_{j+1}}\alpha_{j+1}\ge h_{i_1}\beta_1\sum_{j=0}^{n_i-1}\left(\left(\prod_{k=1}^{j}(1-\frac{\alpha_k}{\beta_k})\right)\frac{\alpha_{j+1}}{\beta_{j+1}}\right)\\&=h_{i_1}\beta_1\left(1-\prod_{j=1}^{n_i}(1-\frac{\alpha_j}{\beta_j})\right)\ge h_{i_1}\beta_1\left(1-\prod_{j=1}^{n_i}(1-\alpha_j)\right),\end{aligned}$$

where the second inequality uses the fact that $\alpha_j\le\beta_j\le 1$, which is true by definition as $t_{\sigma_i(\tau_{j-1})}\le\tau_{j-1}\le\tau_j\le t_{\sigma_i(\tau_{j-1})}+1$.

Moreover, notice that we have $\sum_{j=1}^{n_i}\alpha_j=d_i^*-t_i$, so we can further lower bound the last inequality by considering the maximum possible value for the product expression, which will happen when all of its terms are equal to each other, i.e., $\forall j:\alpha_j=\frac{d_i^*-t_i}{n_i}$. Lastly, by using inequalities $h_{i_1}\beta_1\ge h_i$ and $\forall x: e^x\ge 1+x$, we will get:

$$\int_{t_i}^{d_i^*}h_{\sigma_i(t)}\ dt\ge h_{i_1}\beta_1\left(1-\prod_{j=1}^{n_i}(1-\alpha_j)\right)\ge h_{i_1}\beta_1\left(1-(1-\frac{d_i^*-t_i}{n_i})^{n_i}\right)\ge h_i\left(1-e^{-(d_i^*-t_i)}\right).$$

Finally, the function $\frac{1-e^{-x}}{x}$ is a decreasing function, thereby finishing the proof (as $d_i^*-t_i\le 1$):

$$\int_{t_i}^{d_i^*}h_{\sigma_i(t)}\ dt\ge h_i\left(1-e^{-(d_i^*-t_i)}\right)\ge h_i(d_i^*-t_i)(1-\frac{1}{e}).$$

□

We are now ready to finish the proof of Proposition 2, in which we prove the lower bound of $1-\frac{1}{e}$ as well as provide the tight upper bound (bad instance) for it.

Proof of Proposition 2. Due to the fact that μ_t was a one-to-one correspondence at all time, we are sure that at any point in time t, we do not double count any request that π^{ra} is keeping for different i in $\sigma_i(t)$. As a result, we can write the following:

$$\forall\mathcal{I}:\quad U_\pi^{\mathsf{ra}}(\mathcal{I})=\sum_{\{i\in[N]|t_i<d_i^*\}}\left(\int_{t_i}^{d_i^*}h_{\sigma_i(t)}\ dt\right).$$

Remember that $\mathsf{OPT}(\mathcal{I})=\sum_{i\in[N]}h_i(d_i^*-t_i)=\sum_{\{i\in[N]|t_i<d_i^*\}}h_i(d_i^*-t_i)$, so in order to get any competitive ratio X for π^{ra}, it is sufficient to show that $\forall i\in[N]$ s.t. $t_i<d_i^*$:

$$\frac{\int_{t_i}^{d_i^*}h_{\sigma_i(t)}\ dt}{h_i(d_i^*-t_i)}\ge X.$$

From Lemma EC.4 we know that the inequality holds when $X=1-\frac{1}{e}$, thereby concluding that $\mathcal{CR}(\pi^{\mathsf{ra}})\ge 1-\frac{1}{e}$.

Finally, to see that the competitive ratio of π^{ra} is tight, i.e., $\mathcal{CR}(\pi^{\mathsf{ra}})=1-\frac{1}{e}$, we will provide the following worst-case instance. The instance $\mathcal{I}$ consists of a repeated sequence of continuum arrival of requests. More

specifically, for any round k corresponding to the time interval $[k, k+1)$, we have a continuum of arrivals in the following manner: at time $k+t \in [k, k+1), \forall t \in [0,1)$, we input a request with height e^{-t}, i.e., $z_{t+k} = (t+k, e^{-t})$. Because the gradient of the height function (e^{-t}) for any $t \in (0,1)$ is equal to negative of the height of the request z_{t+k} for any round k, which itself is equal to the gradient of the index function of request z_{t+k}, we can conclude that the policy π^{ra} will accept the request z_{t+k+dt}. As for the requests z_k, i.e., when $t=0$, we know that the algorithm will accept it because $h_{z_k} = 1$, whereas $h_{z_{k^-}} = \frac{1}{e}$ which is the request that π^{h} is serving at the end of the previous round. On the other hand, the optimal offline will only serve requests z_k, and preempts all the rest of requests z_{k+t}, $\forall k \in \mathbb{Z}_+, t \in (0,1)$ immediately as they arrive. Note that if $C > 1$, we can just duplicate each request C times, and all the above argument will still hold. This way $\mathsf{OPT}(\mathcal{I})$ is always serving C requests with reward 1 at each time t (only all the integer requests), while $\pi^{\mathsf{ra}}(\mathcal{I})$ is accepting to serve all requests for infinitesimal amount of time. Thus, the total reward in each interval $[k, k+1), \forall k$ of $\mathsf{OPT}(\mathcal{I})$ would be C, and that of $\pi^{\mathsf{ra}}(\mathcal{I})$ would be equal to $C\int_0^1 e^{-t}dt = C(1-\frac{1}{e})$, which proves the tightness of $1-\frac{1}{e}$ for the competitive ratio, i.e., $\mathcal{CR}(\pi^{\mathsf{ra}}) = 1-\frac{1}{e}$.

□

Appendix D: Proofs of Section 4

D.1. Proofs in Subsection 4.2

For simplicity, we slightly abuse the notation and refer to $\widetilde{\pi^{\mathcal{B}}}$ as just $\tilde{\pi}$, in the rest of this subsection.

D.1.1. Proof of Lemma 2

Proof of Lemma 2. First notice that we have: $U_{\pi^{\mathcal{B}}}(\mathcal{I}) = \sum_{i\in[N]} h_i(d_i - t_i)$, $V_{\tilde{\pi},\mathcal{B}}(\mathcal{I}) := \sum_{i\in[N]} \left(\int_{t_i}^{\tilde{d}_i} \mathcal{B}_{r_{z_i}}(t)\, dt\right)$. Hence, it is sufficient to show that for all $i \in [N]$, we have

$$h_i(d_i - t_i) \geq \gamma_a \left(\int_{t_i}^{\tilde{d}_i} \mathcal{B}_{r_{z_i}}(t)\, dt\right). \tag{EC.3}$$

Now let us consider two cases:

1. $d_i < t_i + 1$: In this case, we have $\tilde{d}_i = d_i$. From the definition of γ_a in (6), if we set $z = z_i$ and $t' = d_i \in [t_i, t_i+1)$, then it follows immediately from the first term in *min* of γ_a:

$$\frac{h_i(d_i - t_i)}{\int_{t_i}^{d_i} \mathcal{B}_{r_{z_i}}(t)\, dt} \geq \gamma_a,$$

which implies (EC.3).

2. $d_i = t_i + 1$: In this case, we have $\tilde{d}_i = d_i + 1 = t_i + 2$. Similarly, from the definition of γ_a in (6), if we set $z = z_i$, then it follows immediately from the second term in *min* of γ_a:

$$\frac{h_i}{\int_{t_i}^{t_i+2} \mathcal{B}_{r_{z_i}}(t)\, dt} \geq \gamma_a,$$

which implies (EC.3).

The proof is completed. □

D.1.2. Proof of Lemma 3

Before going into the proof, let us first state the following remark.

Remark EC.4. We can assume w.l.o.g. that the system is always full under $\pi^{\mathcal{B}}$, as we can add at any time any arbitrary number of dummy requests whose heights are all equal to 0.

Proof of Lemma 3. We can rewrite the LHS and RHS of (8) in the following manner:

- $V_{\tilde{\pi},\mathcal{B}}(\mathcal{I}) := \sum_{i\in[N]} \left(\int_{t_i}^{\tilde{d}_i} \mathcal{B}_{r_{z_i}}(t)\ dt\right) = \int_0^{T+1} \left(\sum_{i\in\mathcal{H}_{\tilde{\pi}}(t)} \mathcal{B}_{r_{z_i}}(t)\right)\ dt \geq \int_0^T \left(\sum_{i\in\mathcal{H}_{\tilde{\pi}}(t)} \mathcal{B}_{r_{z_i}}(t)\right) dt$;
- $V_{\mathsf{OPT},\mathcal{B}}(\mathcal{I}) := \sum_{i\in[N]} \left(\int_{t_i}^{d_i^*} \mathcal{B}_{r_{z_i}}(t)\ dt\right) = \int_0^T \left(\sum_{i\in\mathcal{H}_{\mathsf{OPT}}(t)} \mathcal{B}_{r_{z_i}}(t)\right)\ dt$.

Thus, it is sufficient to show that for all $t\in[0,T)$, we have

$$\sum_{i\in\mathcal{H}_{\tilde{\pi}}(t)} \mathcal{B}_{r_{z_i}}(t) \geq \sum_{i\in\mathcal{H}_{\mathsf{OPT}}(t)} \mathcal{B}_{r_{z_i}}(t). \tag{EC.4}$$

Claim EC.3. *Given any time t and any request z_k, if we have*

$$\min_{i\in\mathcal{H}_{\pi^{\mathcal{B}}}(t)} \left\{\mathcal{B}_{r_{z_i}}(t)\right\} \geq \mathcal{B}_{r_{z_k}}(t), \tag{EC.5}$$

then we also have

$$\forall t'\in[t,t+1): \min_{i\in\arg\max^C\left\{\mathcal{B}_{r_{z_j}}\,\middle|\, j\in\mathcal{H}_{\tilde{\pi}}(t')\right\}} \left\{\mathcal{B}_{r_{z_i}}(t')\right\} \geq \mathcal{B}_{r_{z_k}}(t').$$

Proof of Claim EC.3. The following fact comes from (5):

$$\exists t \text{ such that } \mathcal{B}_{r_{z_i}}(t) \geq \mathcal{B}_{r_{z_k}}(t) \iff r_{z_i} \geq r_{z_k} \iff \mathcal{B}_{r_{z_i}}(t) \geq \mathcal{B}_{r_{z_k}}(t)\ \forall t.$$

Combining the above with the assumption in (EC.5), we get:

$$\min_{i\in\mathcal{H}_{\pi^{\mathcal{B}}}(t)} \{r_{z_i}\} \geq r_{z_k}.$$

Now notice that from the behavior of the index-based policy $\pi^{\mathcal{B}}$ we know that, unless any of its requests hit the end of its lifespan and the policy is forced to preempt that request, $\pi^{\mathcal{B}}$ preempts a request k at some time t only if there is a newly-arrived request i whose index at the time is higher than that of request k's, i.e., $f_{z_i}(t) \geq f_{z_k}(t)$ - which itself is equivalent to $\mathcal{B}_{r_{z_i}}(t) \geq \mathcal{B}_{r_{z_k}}(t)$, or simply $r_{z_i} \geq r_{z_k}$.

Now let us prove the claim. Fix an arbitrary t and $t'\in[t,t+1)$, define $D[t,t']$ as the set of the requests that $\pi^{\mathcal{B}}$ removes in the time interval $[t,t']$ due to the reason they have hit the end of their lifespans. From the way we defined $\tilde{\pi}$ we know that it will serve all these requests for an extra unit of time after their deadline. From this we can deduce that because $t'\in[t,t+1)$, then all of $D[t,t']$ would be still being served by $\tilde{\pi}$ at time t', i.e., $D[t,t']\subseteq\mathcal{H}_{\tilde{\pi}}(t')$.

Let us first define relation $\preceq$ between two sets A and B. We say $A\preceq B$, if $|A|\leq|B|$ and if we rank the requests in each of them based on the requests' r_z, then $\forall l\in[|A|]$, the $l-th$ highest r_z in A is no greater than the $l-th$ highest r_z in B.

Now consider $\mathcal{H}_{\pi^{\mathcal{B}}}(t)$. This set consists of C number of requests. Until time t' each of those requests might experience 3 types of fate: i) remain in the system, ii) might get replace multiple times (without ever - the original request or any of its replacement chain - hitting their deadline), and iii) might hit their deadline after some (possibly 0) number of replacements (and probably get replaced by a zero-reward request). Let us name the set of requests in each of these fate classes as F_1,F_2,F_3, respectively. First, by definition of $\pi^{\mathcal{B}}$, we have $\forall s:\mathcal{H}_{\pi^{\mathcal{B}}}(s)\subseteq\mathcal{H}_{\tilde{\pi}}(s)$. Therefore, we immediately get: $F_1\subseteq\mathcal{H}_{\pi^{\mathcal{B}}}(t')\subseteq\mathcal{H}_{\tilde{\pi}}(t')$. Second, we get: $F_3\preceq D[t,t']\subseteq\mathcal{H}_{\tilde{\pi}}(t')\backslash\mathcal{H}_{\pi^{\mathcal{B}}}(t')$, because of the fact that, in each of the replacement chains, the r_z sequence is increasing, and the final element of the chain is indeed in $D[t,t']$ by definition. Lastly, as for the second type,

F_2, again each of the chains will have an increasing sequence of r_z, and for this type, the chain is still alive by time t', meaning the elements of these chains at time t' is also an element of $\mathcal{H}_{\tilde{\pi}}(t')\backslash\{\mathcal{H}_{\pi^{\mathcal{B}}}(t')\cup D[t,t']\}$. Therefore, we get $F_2 \preceq \mathcal{H}_{\tilde{\pi}}(t')\backslash\{\mathcal{H}_{\pi^{\mathcal{B}}}(t')\cup D[t,t']\}$.

Combining all these three pieces together, we can get: $\mathcal{H}_{\pi^{\mathcal{B}}}(t) \preceq \mathcal{H}_{\tilde{\pi}}(t')$, and because size of $\mathcal{H}_{\pi^{\mathcal{B}}}(t)$ was C, then together with that "equivalent" assumption of the claim we can get:

$$r_{z_k} \le \min_{i\in\mathcal{H}_{\pi^{\mathcal{B}}}(t)}\{r_{z_i}\} \le \min_{i\in\arg\max^C\left\{\mathcal{B}_{r_{z_j}}\,\middle|\, j\in\mathcal{H}_{\tilde{\pi}}(t')\right\}}\{r_{z_i}\},$$

which implies

$$\min_{i\in\arg\max^C\left\{\mathcal{B}_{r_{z_j}}\,\middle|\, j\in\mathcal{H}_{\tilde{\pi}}(t')\right\}}\{\mathcal{B}_{r_{z_i}}(t')\} \ge \mathcal{B}_{r_{z_k}}(t').$$

The proof of the claim is completed. □

Now let us prove equation (EC.4) with the help of Claim EC.3 for any arbitrary time t.

We divide $\mathcal{H}_{\mathsf{OPT}}(t)$ into two subset, $\mathcal{J} \triangleq \mathcal{H}_{\mathsf{OPT}}(t)\cap\mathcal{H}_{\pi^{\mathcal{B}}}(t)$ and $\mathcal{D} \triangleq \mathcal{H}_{\mathsf{OPT}}(t)\setminus\mathcal{H}_{\pi^{\mathcal{B}}}(t)$. Now notice for any $k\in\mathcal{D}$, we know that the policy $\pi^{\mathcal{B}}$ must have preempted that request sometime along the way before time t, i.e., $d_k < t$. From the fact that request k is being kept by OPT until time t we conclude that request k had not reached its deadline at time d_k, meaning that the $\pi^{\mathcal{B}}$ had intentionally preempted request k at time d_k. This means that the indices of all requests that $\pi^{\mathcal{B}}$ was going to serve at time d_k must have been all at least as high as the index of request k at that time d_k; in other words,

$$\min_{i\in\mathcal{H}_{\pi^{\mathcal{B}}}(d_k)}\{\mathcal{B}_{r_{z_i}}(d_k)\} \ge \mathcal{B}_{r_{z_k}}(d_k).$$

Next we will use the claim, where you replace t,t' in the claim by d_k,t respectively. Now from the fact that the request k must have been arrived as late as its departure time by $\pi^{\mathcal{B}}$ and its deadline must have been as early as its departure time by OPT, i.e., $t_k \le d_k \le t \le t+1$, we conclude that $t\in[d_k, d_k+1)$. Combining the above with (D.1.2), we see that the assumption in the claim holds. Therefore,

$$\min_{i\in\arg\max^C\left\{\mathcal{B}_{r_{z_j}}\,\middle|\, j\in\mathcal{H}_{\tilde{\pi}}(t)\right\}}\{\mathcal{B}_{r_{z_i}}(t)\} \ge \mathcal{B}_{r_{z_k}}(t).$$

Finally, notice that we have showed the above inequality $\forall k\in\mathcal{D}$. Moreover, notice that $|\mathcal{H}_{\mathsf{OPT}}(t)|\le C$ as well as $\mathcal{J}\subseteq\mathcal{H}_{\pi^{\mathcal{B}}}(t)\subseteq\mathcal{H}_{\tilde{\pi}}(t)$. It follows that

$$\mathcal{H}_{\mathsf{OPT}}(t)\preceq\mathcal{H}_{\tilde{\pi}}(t) \implies \sum_{i\in\mathcal{H}_{\tilde{\pi}}(t)}\mathcal{B}_{r_{z_i}}(t) \ge \sum_{i\in\mathcal{H}_{\mathsf{OPT}}(t)}\mathcal{B}_{r_{z_i}}(t).$$

The proof of Lemma 3 is completed.

□

D.1.3. Proof of Lemma 4

Proof of Lemma 4. The argument is similar to the one of Lemma 2. First, notice that we have: $\mathsf{OPT}(\mathcal{I}) = \sum_{i\in[N]} h_i(d_i^* - t_i)$, $V_{\mathsf{OPT},\mathcal{B}}(\mathcal{I}) := \sum_{i\in[N]} \left(\int_{t_i}^{d_i^*} \mathcal{B}_{r_{z_i}}(t)\ dt\right)$. Hence, it is sufficient to show that for all $i \in [N]$, we have

$$\left(\int_{t_i}^{d_i^*} \mathcal{B}_{r_{z_i}}(t)\ dt\right) \geq \gamma_o h_i(d_i^* - t_i).$$

This inequality itself is again a direct consequence of the definition of γ_o. If we set $z = z_i$ and $t' = d_i^* \in [t_i, t_i + 1)$, then it follows immediately from the definition of γ_o in (7):

$$\frac{\int_{t_i}^{d_i^*} \mathcal{B}_{r_{z_i}}(t)\ dt}{h_i(d_i^* - t_i)} \geq \gamma_o,$$

completing the proof.

□

D.1.4. Proof of Theorem 1

Proof of Theorem 1. To prove Theorem 1, we combine the result of all three Lemmas 2 to 4 together, and we can get what we want:

$$U_{\pi^{\mathcal{B}}}(\mathcal{I}) \geq \gamma_a \sum_{i\in[N]} \left(\int_{t_i}^{\tilde{d}_i} \mathcal{B}_{r_{z_i}}(t)\ dt\right) \geq \gamma_a \sum_{i\in[N]} \left(\int_{t_i}^{d_i^*} \mathcal{B}_{r_{z_i}}(t)\ dt\right) \geq \gamma_a\gamma_o \mathsf{OPT}(\mathcal{I}).$$

□

D.2. Proofs in Subsection 4.3

D.2.1. Proof of Lemma 5

Proof of Lemma 5. Please refer to Section 7 and Lemma 7 for the derivation of γ_a and γ_o under a more generalized setting, in which the reward rate of each request can vary over time. This setting is a special case of that when $R = 1$ (refer to there for definition of $\mathfrak{r}(.)$ and R), and the height of each request is a constant function (meaning $\mathfrak{r}(.) = 1$) which clearly satisfies both homogeneity and non-increasing assumptions needed in Lemma 7. □

D.2.2. Proof of Theorem 2

Proof of Theorem 2. The lower bound part, i.e., $\mathcal{CR}(\pi^{\exp(\alpha^*)}) \geq \frac{1-e^{-\alpha^*}}{\alpha^*} \geq 0.68$ can be readily achieved by combining Theorem 1 and Lemma 5. The rest is to show that this ratio is, indeed, tight. We prove this by providing a bad instance $\mathcal{I}$ under which the performance ratio of $\pi^{\exp(\alpha^*)}(\mathcal{I})$ against $\mathsf{OPT}(\mathcal{I})$ is, in fact, equal to $\frac{1-e^{-\alpha^*}}{\alpha^*}$. Instance $\mathcal{I}$ consists of only 2 type of requests each duplicated C times, i.e., $z_1 := (t_1 = 0, h_1 = 1)$ and $z_2 := (t_2 = 1^-, h_2 = e^{-\alpha^*})$ where we have C duplicate of each of them. Under this instance, the policy will preempt all of the second type requests immediately and just keep serving all first types till their deadline, $t = 1$. However, the optimal offline policy will preempt all the first type requests when the second type ones arrive and start serving them until the end of their deadline which is $t = 2^-$. As a result we will have $U_{\pi^{\exp(\alpha^*)}}(\mathcal{I}) = C$, while $\mathsf{OPT}(\mathcal{I}) = C + Ce^{-\alpha^*} = C(1 + e^{-\alpha^*})$. Therefore, their ratio will be:

$$\mathcal{CR}(\pi^{\exp(\alpha^*)}) \leq \frac{U_{\pi^{\exp(\alpha^*)}}(\mathcal{I})}{\mathsf{OPT}(\mathcal{I})} = \frac{1}{1+e^{-\alpha^*}} = \frac{1-e^{-\alpha^*}}{\alpha^*}.$$

Note that the last equality comes from the choice of α^*, i.e., $1 - \alpha^* = e^{-2\alpha^*} \to \alpha^* = 1 - e^{-2\alpha^*} = (1 - e^{-\alpha^*})(1 + e^{-\alpha^*})$. This concludes the tightness of the competitive ratio of this policy, i.e., $\mathcal{CR}(\pi^{\exp(\alpha^*)}) = \frac{1-e^{-\alpha^*}}{\alpha^*}$.

□

Appendix E: Proofs in Section 5

E.1. Proof of Proposition 3

Proof of Proposition 3. Consider the continuum of arrival of requests over the time interval $[0,1]$, where at each time t we input C duplicates of request $z_t := (t, h_t := 1-t), \forall t \in [0,1]$. It is not hard to see that an optimal policy is to serve all the C duplicates of the request z_0 until the end of their lifespan, i.e., $t=1$. Therefore, we get $\mathsf{OPT} = C$. On the other hand, Skyline not only will keep serving all C duplicates of the z_0 until $t=1$, but also it will serve all the departed requests for an infinitesimal duration of time at their deadline. As a result, we would have $\mathsf{Skyline} = C(1+\int_1^2 h_{t-1}dt) = C\frac{3}{2}$. This will tell us that there is at least a gap of $\frac{2}{3}$ on the ratio of OPT over Skyline under worst-case instance, so as on the ratio of any policy π against the Skyline. □

E.2. Proof of Proposition 4

Proof of Proposition 4. We prove this proposition by providing a strategy for the adversary that any feasible deterministic online algorithm will perform poorly under that. In particular, given any feasible deterministic online algorithm π, adversarial instance $\mathcal{I}^{Adv}$ is constructed in the following manner. Suppose the first request arrives with $z_1 := (t_1 = 0, h_1 = 1)$. After that, the second request will arrive with parameters $z_2 := (t = x^*, h = x^*)$, where $x^* \approx 0.597$ is the unique solution to $\frac{1}{2x} = \frac{x(2-x)}{1}$ on $[0,1]$. At time x^* any algorithm needs to decide which one of the request 1 or 2 should be preempted from the system. If π decided to preempt request 2 and continue on serving request 1, the adversary will cut any further arrival to the system, i.e., make request 2 the last arrived request. On the other hand, if the policy decided to preempt request 1 and start serving request 2, then at time $t=1$, the adversary will restart the instance and repeat this process for a new unit time, i.e., it assumes $t=0$, and it will input to the system the two requests 1 and 2 as specified earlier.

Now we claim that under instance $\mathcal{I}^{Adv}$ - no matter how many rounds are passed based on the response of the algorithm, where in each round k the adversary inputs a pair of requests (request $z_{2k-1} := (k-1, 1)$ and request $z_{2k} := (k-1+x^*, x^*)$) - the ratio of the total reward that π has accumulated compared to that of the optimal offline would be equal to $x^*(2-x^*)$.

To prove this claim, suppose a total of K rounds have passed until the instance $\mathcal{I}^{Adv}$ got terminated by the adversary. This means that the algorithm has accepted to serve all new arrivals until the arrival of the next request, except for the last request $2K$ for which the algorithm has chosen to preempt it immediately and continued serving the request $2K-1$ until its deadline $t_{2K-1}+1 = K$.

As for the optimal offline, it will only serve the odd requests and preempts all even requests immediately, except the last request $2K$. More specifically, all requests $2i-1$ for all $i \in [K-1]$ will be served in the system for a whole unit of time, request $2K-1$ will be served for period with length x^*, and lastly, request $2K$ will be served for a whole unit of time again.

Accordingly, the utility that algorithm accumulates in each round $i \in [K-1]$ (the time interval $[i-1, i)$) is equal to $x^*(2-x^*)$, while it accumulated a total reward of 1 from the request $2K-1$, and nothing more. On the other hand, the optimal offline is accumulating a total reward of 1 in each of the rounds $i \in [K-1]$, as well as a total of x^* from each of the requests $2K-1$ and $2K$.

Finally, we get that the ratio of the reward of the algorithm to optimal offline in each of the rounds $i \in [K-1]$ is $\frac{x^*(2-x^*)}{1}$, while the ratio of their total reward after time $t = K-1$ would be $\frac{1}{2x^*}$. Now due to the fact that x^* satisfies $\frac{1}{2x^*} = \frac{x^*(2-x^*)}{1}$, we can conclude that the ratio of total reward of the algorithm compared to that of the optimal offline under such instance $\mathcal{I}^{Adv}$ is $x^*(2-x^*)$, concluding the proof of the claim. Consequently, our proposition is proved from the fact that the competitive ratio is, in fact, the minimum ratio under such adversarial instances (strategies). □

E.3. Proof of Proposition 5

Proof of Proposition 5. We will provide a mixed strategy for the oblivious adversary. Recall the strategy the adversary was using in the proof of Proposition 4 in Subsection E.2. It was a repeated round instance, where in each round k, the adversary was inputting the two requests $z_1 := (t_1 = k-1, h_1 = 1)$ and $z_2 := (t = k-1+x^*, h = x^*)$, where $x^* \approx 0.597$ was the solution to $\frac{1}{2x} = \frac{x(2-x)}{1}$. In there, the stopping time K (end of the last round K when the adversary terminates sending any new round after that) was computed based on the decisions of the policy π, which was deterministic.

Now that the policy π can be a randomized policy, we consider the instance $\mathcal{I}$ in which the adversary will input the same pair of requests in each round k - with parameters exactly as described in the deterministic version in Subsection E.2 - with probability $\frac{1}{2}$ independently at random, and with probability $\frac{1}{2}$, the adversary skips feeding the pair in round k and awaits until the time of next round $k+1$ to arrive and repeats this coin flipping process forever. Under such a random instance, it is not hard to see that at any time k (the end of round k), if the next round would be skipped then the optimal algorithm will continue serving the request it had from round k until its deadline, and if the round $k+1$ arrives, then it is optimal to switch to the fresh z_1 (type-1) request of the new round.

With this in mind, we can see that at any time $k-1+x^*$ (the time in round k that the algorithm should decide whether or not to switch from serving the type-1 request z_1 to the type-2 request z_2) no matter what decision it takes, the competitive ratio of its decision (corresponding to the total reward achieved only from the requests in round k) will be the same and equal to $\frac{1+x^*(2-x^*)}{2} < \frac{1+0.8376}{2} \approx 0.92$, and for the next non-skipped rounds everything will be reset due to the fact that we showed any optimal policy will switch at the beginning of a non-skipped round.

Accordingly, with this mixed strategy, the adversary has made any feasible online policy π to be indifferent in the middle-round decisions. Thus, showing us that the competitive ratio of any feasible randomized online policy would be now upper bounded by $\frac{1+x^*(2-x^*)}{2} < \frac{1+0.8376}{2} \approx 0.92$. □

Appendix F: Proofs of Section 6

F.1. Proofs in Subsection 6.1

F.1.1. Proof of Theorem 3

Proof of Theorem 3. We prove that the following chain of inequalities holds:

$$U_{\pi^{\mathsf{red}}}(\mathcal{I}) \geq U_{\widetilde{\pi}}(\widetilde{\mathcal{I}}) \geq \widetilde{\mathcal{CR}} \times \mathsf{OPT}(\widetilde{\mathcal{I}}) \geq \frac{\widetilde{\mathcal{CR}}}{\sqrt{L}} \times \mathsf{OPT}(\mathcal{I}).$$

To do so, let us prove each of the three inequalities one by one:

1. $U_{\pi^{\mathsf{red}}}(\mathcal{I}) \geq U_{\widetilde{\pi}}(\widetilde{\mathcal{I}})$: We prove that for any request i, the total reward corresponding to it that π^{red} acquires is higher than the amount that $\widetilde{\pi}$ acquires. To see this, we consider all possible cases in the following:

(a) $\ell_i \geq \sqrt{L}$: In this case, we know $d_i^{\pi^{\mathsf{red}}} = d_i^{\widetilde{\pi}}$, and the arrival time and the reward rate for such a request were the same under both $\widetilde{\mathcal{I}}$ and $\mathcal{I}$, hence they both receive exactly the same reward from request i.

(b) $\ell_i < \sqrt{L}$, $d_i^{\widetilde{\pi}} \leq t_i + \ell_i$: In this case, we again know $d_i^{\pi^{\mathsf{red}}} = d_i^{\widetilde{\pi}}$. Additionally, notice that we had $h_i > \widetilde{h}_i$; therefore, we can again get $h_i(d_i^{\pi^{\mathsf{red}}} - t_i) \geq \widetilde{h}_i(d_i^{\widetilde{\pi}} - t_i)$.

(c) $\ell_i < \sqrt{L}$, $d_i^{\widetilde{\pi}} > t_i + \ell_i$: In this case, we have $d_i^{\pi^{\mathsf{red}}} = t_i + \ell_i$. As a result, we get $h_i(d_i^{\pi^{\mathsf{red}}} - t_i) = h_i\ell_i = \widetilde{h}_i\widetilde{\ell}_i \geq \widetilde{h}_i(d_i^{\widetilde{\pi}} - t_i)$, which is again what we wanted to prove.

2. $U_{\widetilde{\pi}}(\widetilde{\mathcal{I}}) \geq \widetilde{\mathcal{CR}} \times \mathsf{OPT}(\widetilde{\mathcal{I}})$: Proof of this part is just by definition, i.e., $\widetilde{\mathcal{CR}}$ was, in fact, the competitive ratio of $\widetilde{\pi}$ under the homogeneous lifespan setting, which subsumes $\widetilde{\mathcal{I}}$.

3. $\mathsf{OPT}(\widetilde{\mathcal{I}}) \geq \frac{1}{\sqrt{L}} \times \mathsf{OPT}(\mathcal{I})$: In order to prove this last inequality, we will provide a feasible policy π under the homogeneous instance $\widetilde{\mathcal{I}}$ whose utility satisfies: $\mathsf{OPT}(\widetilde{\mathcal{I}}) \geq U_\pi(\widetilde{\mathcal{I}}) \geq \frac{1}{\sqrt{L}} \times \mathsf{OPT}(\mathcal{I})$. Note that the first inequality automatically follows by definition of optimality of $\mathsf{OPT}(\widetilde{\mathcal{I}})$ and feasibility of π under $\widetilde{\mathcal{I}}$. To define behavior of π under $\widetilde{\mathcal{I}}$ uniquely, again we only need to determine its departure times for all request in $\widetilde{\mathcal{I}}$. Thus, we set $d_i^\pi = \min(d_i^{\mathsf{OPT}(\mathcal{I})}, t_i + \widetilde{\ell}_i)$. By this way of defining the departure time as well as the fact that $\mathsf{OPT}(\mathcal{I})$ was feasible under $\mathcal{I}$ (w.r.t. capacity and lifespan constraints), we can easily see that π is feasible under $\widetilde{\mathcal{I}}$. The last remaining part to show is the inequality $U_\pi(\widetilde{\mathcal{I}}) \geq \frac{1}{\sqrt{L}} \times \mathsf{OPT}(\mathcal{I})$. Similarly, we will show that for any request i, the total amount of reward corresponding to i under both $\pi(\widetilde{\mathcal{I}})$ and $\mathsf{OPT}(\mathcal{I})$, satisfies the inequality. Consider the following two possible cases:

(a) $\ell_i \geq \sqrt{L}$: For this class of requests, we have $\widetilde{h}_i = h_i$ and $d_i^\pi - t_i = \min(d_i^{\mathsf{OPT}(\mathcal{I})} - t_i, \sqrt{L}) \geq \frac{1}{\sqrt{L}}(d_i^{\mathsf{OPT}(\mathcal{I})} - t_i)$, because $d_i^{\mathsf{OPT}(\mathcal{I})} - t_i \leq L$. As a consequence, we get $\widetilde{h}_i(d_i^\pi - t_i) \geq h_i \frac{1}{\sqrt{L}}(d_i^{\mathsf{OPT}(\mathcal{I})} - t_i)$, which is exactly what we wanted to show.

(b) $\ell_i < \sqrt{L}$: For this other class of requests, we have $d_i^\pi = d_i^{\mathsf{OPT}(\mathcal{I})}$ and $\widetilde{h}_i = \frac{h_i\ell_i}{\sqrt{L}} \geq \frac{h_i}{\sqrt{L}}$, because $\ell_i \geq 1$. As a result, we get $\widetilde{h}_i(d_i^\pi - t_i) \geq \frac{h_i}{\sqrt{L}}(d_i^{\mathsf{OPT}(\mathcal{I})} - t_i)$, which is again what we wanted to show.

Putting all these three pieces of chain inequalities together, we conclude the proof of the theorem.

□

F.1.2. Proof of Proposition 6

Proof of Proposition 6. It is enough to provide an adversary instance generation strategy under which no deterministic feasible online algorithm is able to beat $\frac{1}{\sqrt{L}}$ fraction of the OPT utility. The bad instance $\mathcal{I}$ will be generated as follows. Fix a deterministic feasible online algorithm π. At time $t = 0$, the adversary will simultaneously input two requests $z_1 = (t_1 = 0, h_1 = \sqrt{L}, \ell_1 = 1)$ and $z_2 = (t_2 = 0, h_2 = 1, \ell_2 = L)$. If π chooses to serve z_1, then the adversary will terminate the instance, i.e., will send no more arrivals to the system. Otherwise, i.e., if π chooses to serve z_2, then at time $t = 1$, the adversary imagines $t = 0$ and repeat the entire pair arrival again, with two new requests z_3 and z_4 that are defined in exactly the same way as above, except for their arrival time which is now 1 for both of them. The adversary will continue on inputting these pairs of requests (at integer times) until the first time, say at round k, that the algorithm decides to serve the request z_{2k-1} and preempt request z_{2k}, after which the adversary will send no more arrivals to the system.

To see that why the utility ratio of the algorithm against optimal offline under such an instance is at most $\frac{1}{\sqrt{L}}$, suppose the instance consists of a total of K repeated rounds before it gets terminated. The algorithm must have chosen to serve all even requests $z_{2k}, \forall k \in [K-1]$ as well as request z_{2K-1}, while the optimal offline - under such instance - will choose to serve all odd requests $z_{2k-1}, \forall k \in [K-1]$ together with the last request z_{2K}. Thus, the utility of the algorithm $U_\pi(\mathcal{I}) = (K-1) + \sqrt{L}$, whereas that of the optimal offline would be $\mathsf{OPT}(\mathcal{I}) = (K-1)\sqrt{L} + L$, which generates a competitive ratio of at most $\mathcal{CR}(\pi) \leq \frac{1}{\sqrt{L}}$ and concludes our proof. □

F.2. Proofs in Subsection 6.2

F.2.1. Proof of Lemma 6

Proof of Lemma 6. The proof of the lemma concludes from the following chain of inequalities:

$$\begin{aligned} U_{\pi_K^{\mathsf{div}}}(\mathcal{I}; C) &= \sum_{k\in[K]} U_{\pi^k}\left(\mathcal{I}_k; \lfloor \frac{C}{K} \rfloor\right) \\ &\geq \sum_{k\in[K]} \frac{\widetilde{\mathcal{CR}}}{\sqrt{L^{\frac{1}{K}}}} \mathsf{OPT}\left(\mathcal{I}_k; \lfloor \frac{C}{K} \rfloor\right) \\ &\geq \frac{\widetilde{\mathcal{CR}}}{\sqrt{L^{\frac{1}{K}}}} \sum_{k\in[K]} \frac{\lfloor \frac{C}{K} \rfloor}{C} \mathsf{OPT}(\mathcal{I}_k; C) \\ &\geq \frac{\widetilde{\mathcal{CR}}}{\sqrt{L^{\frac{1}{K}}}} \frac{\lfloor \frac{C}{K} \rfloor}{C} \mathsf{OPT}(\mathcal{I}; C). \end{aligned}$$

Before delving into the rationale behind each inequality, it is essential to recall that the second argument (C or $\lfloor \frac{C}{K} \rfloor$) in the utility functions ($U_\pi(\mathcal{I}; C)$ or $\mathsf{OPT}(\mathcal{I}; C)$) represents the system capacity under which the algorithm operates. For the sake of brevity, we had omitted this parameter so far, as our analysis focused solely on a single value of capacity. However, with the introduction of distinct capacities for our sub-problems compared to the original problem, it becomes necessary to reintroduce this parameter for clarity. Now let us state the reasoning behind each step of this chain of inequalities:

1. The first equality holds because π_K^{div} is exactly emulating all of the π^k, $\forall k \in [K]$, simultaneously.

2. The second inequality holds from the fact that the competitive ratio of each π^k (with capacity $\lfloor \frac{C}{K} \rfloor$) under $\mathcal{I}_k$ was $\frac{\widetilde{\mathcal{CR}}}{\sqrt{L^{\frac{1}{K}}}}$.

3. The third inequality holds from the simple observation that if you multiply the capacity of OPT by some factor, its utility will be increased by at most that factor.

4. The fourth inequality results from the following two facts. First, $\mathcal{I} = \cup_{k\in[K]} \mathcal{I}_k$. Second, for any $k \in [K]$, the way $\mathsf{OPT}(\mathcal{I}; C)$ serves just the requests in $\mathcal{I}_k$, would itself be a feasible policy under the instance $\mathcal{I}_k$ when capacity is still C. Therefore, the share of the utility $\mathsf{OPT}(\mathcal{I}; C)$ that comes from the requests in $\mathcal{I}_k$ is less than the $\mathsf{OPT}(\mathcal{I}_k; C)$. Finally, by taking the summation over all categories, we get the inequality we want, since from the first fact we are sure that we have accounted for each request's share in $\mathsf{OPT}(\mathcal{I}_k; C)$ at least once.

□

F.2.2. Proof of Theorem 4

Proof of Theorem 4. From Lemma 6, we know that

$$\mathcal{CR}\left(\pi^{\mathsf{div}}_{K^*}\right) \geq \max_{K\in[C]} \left\{ \frac{\widetilde{\mathcal{CR}}}{\sqrt{L^{\frac{1}{K}}}} \frac{\lfloor \frac{C}{K} \rfloor}{C} \right\}.$$

Hence, it is sufficient to show that whenever $C = \Omega(\log(L))$, we have

$$\max_{K\in[C]} \left\{ \frac{1}{\sqrt{L^{\frac{1}{K}}}} \frac{\lfloor \frac{C}{K} \rfloor}{C} \right\} = \Omega\left(\frac{1}{\log(L)}\right).$$

Let w be a small constant such that $C \geq w\log(L)$. Now let $v \ll w$ also be a constant, which is much smaller than w, say $\frac{w}{v} > 11$. Now let $\hat{K} := \lceil v\log(L)\rceil < \frac{w\log(L)}{10} \leq \frac{C}{10}$, where L is sufficiently large ($v\log(L) > 10$). As a consequence, we will get

$$\frac{1}{\sqrt{L^{\frac{1}{K}}}} \geq \frac{1}{e^{\frac{1}{2v}}} = \Omega(1), \quad \frac{\lfloor \frac{C}{\hat{K}} \rfloor}{C} \geq \frac{\frac{C}{\hat{K}} - 1}{C} = \left(\frac{1}{\hat{K}} - \frac{1}{C}\right) = \frac{0.9}{\hat{K}} + \left(\frac{1}{10\hat{K}} - \frac{1}{C}\right) \geq \frac{0.9}{\hat{K}} = \Omega\left(\frac{1}{\log(L)}\right).$$

Thus, we get $\max_{K\in[C]} \left\{ \frac{1}{\sqrt{L^{\frac{1}{K}}}} \frac{\lfloor \frac{C}{K} \rfloor}{C} \right\} \geq \frac{1}{\sqrt{L^{\frac{1}{\hat{K}}}}} \frac{\lfloor \frac{C}{\hat{K}} \rfloor}{C} = \Omega\left(\frac{1}{\log(L)}\right)$, completing the proof of the theorem. □

Appendix G: Proofs in Section 7

G.1. Proof of Lemma 7

Proof of Lemma 7. The two equalities are simply from assumption (11) and how we define r_z, which equals $\mathfrak{h}(t)e^{\alpha t}$. The two inequalities can be derived by setting $t' \uparrow t+1$ and $t' \downarrow t$, respectively. Lastly, if the function $\frac{e^{-\alpha t''}}{\mathfrak{r}(t'')}$ is non-increasing in t'' on $[0,1]$, then we claim that

$$\frac{\int_0^{t'-t} e^{-\alpha t''}\, dt''}{\int_0^{t'-t} \mathfrak{r}(t'')\, dt''} \tag{EC.6}$$

is also non-increasing in t' on $[t, t+1]$. This is due to the fact that if a ratio of two functions $\frac{a(t)}{b(t)}$ is non-increasing in t and $b(t) > 0$ for all $t \geq 0$, then $\frac{\int_0^T a(t)dt}{\int_0^T b(t)dt}$ would also be non-increasing in T for all $T \geq 0$. To see this, note that

$$\frac{d\left(\frac{\int_0^T a(t)dt}{\int_0^T b(t)dt}\right)}{dT} = \frac{a(T)\int_0^T b(t)dt - b(T)\int_0^T a(t)dt}{\left(\int_0^T b(t)dt\right)^2} = \frac{b(T)}{\int_0^T b(t)dt}\left[\frac{a(T)}{b(T)} - \frac{\int_0^T a(t)dt}{\int_0^T b(t)dt}\right],$$

where

$$\frac{\int_0^T a(t)dt}{\int_0^T b(t)dt} = \frac{\int_0^T b(t)\frac{a(t)}{b(t)}dt}{\int_0^T b(t)dt} \geq \frac{\int_0^T b(t)\frac{a(T)}{b(T)}dt}{\int_0^T b(t)dt} = \frac{a(T)}{b(T)},$$

which implies $\frac{d\left(\frac{\int_0^T a(t)dt}{\int_0^T b(t)dt}\right)}{dT} \leq 0$. The monotonicity of (EC.6) implies that the minimum value for each of γ_o and γ_a's min operator is, in fact, achieved when $t' \uparrow t+1$ and $t' \downarrow t$, respectively, thereby making the inequalities hold with equality and completing the proof of the lemma. □