Keep-Alive Caching for Hawkes Processes (Supplementary Material)

Sushirdeep Narayana[1] **Ian A. Kash**[1]

snaray25@uic.edu[1] **iankash@uic.edu**[1]

[1]Department of Computer Science,, University of Illinois at Chicago,, Chicago, Illinois, USA

This Supplementary Material contains proofs and other material omitted from the main manuscript.

A OMITTED PROOFS

Lemma 1. *The expected cost of a cache policy over an inter-arrival is*

$$\mathbb{E}[cost(\pi(\cdot|\mathcal{H}_{m-1}))] = c_{cs} + \int_0^\infty \pi(x|\mathcal{H}_{m-1}) \cdot g(x|\mathcal{H}_{m-1})\, dx,$$

where the instantaneous cost at x units after the most recent arrival at t_{m-1} is

$$g(x|\mathcal{H}_{m-1}) = c_p \cdot \big(1 - F(x|\mathcal{H}_{m-1})\big) - c_{cs} \cdot f(x|\mathcal{H}_{m-1}).$$

Proof. Let $\mathcal{L} = \{L_0, L_1, L_2, \cdots, L_{2k-1}\}$ denote the set of points on the sequence of keep-alive windows for policy $\pi(\cdot|\mathcal{H}_{m-1})$ where even indices are the start of the windows and odd indices are the endpoints of the windows. Let $Z(\mathcal{L}, j) = \sum_{p=0}^{j}(L_{2p+1} - L_{2p})$ for $j \geq 0$. The function $Z(\mathcal{L}, j)$ represents the time accumulated in the cache through the j-th sequence of the keep-alive window. For $j < 0$, we have $Z(\mathcal{L}, j) = 0$. Then we have

$$\begin{aligned}
&\mathbb{E}[cost(\pi(\cdot|\mathcal{H}_{m-1}))] \\
&= c_{cs} \cdot \int_0^{L_0} f(x|\mathcal{H}_{m-1})\, dx \; + \sum_{j=0}^{k-1} \int_{L_{2j}}^{L_{2j+1}} c_p \cdot \Big(Z(\mathcal{L}, j-1) + x - L_{2j}\Big) \cdot f(x|\mathcal{H}_{m-1})\, dx \\
&\quad + \sum_{j=0}^{k-2} \int_{L_{2j+1}}^{L_{2j+2}} \Big(c_{cs} + c_p Z(\mathcal{L}, j)\Big) f(x|\mathcal{H}_{m-1})\, dx \; + \int_{L_{2k-1}}^{\infty} \Big(c_{cs} + c_p Z(\mathcal{L}, k-1)\Big) f(x|\mathcal{H}_{m-1})\, dx \quad (1) \\
&= c_{cs} \cdot F(L_0|\mathcal{H}_{m-1}) + \sum_{j=0}^{k-1} \left(c_p \cdot \Big(Z(\mathcal{L}, j-1) + x - L_{2j}\Big) \cdot F(x|\mathcal{H}_{m-1})\Big|_{L_{2j}}^{L_{2j+1}} - \int_{L_{2j}}^{L_{2j+1}} c_p F(x|\mathcal{H}_{m-1})\, dx \right) \\
&\quad + \sum_{j=0}^{k-2} \Big(c_{cs} + c_p \cdot Z(\mathcal{L}, j)\Big) \cdot F(x|\mathcal{H}_{m-1})\Big|_{L_{2j+1}}^{L_{2j+2}} + \Big(c_{cs} + c_p \cdot Z(\mathcal{L}, k-1)\Big) \cdot F(x|\mathcal{H}_{m-1})\Big|_{L_{2k-1}}^{\infty} \quad (2)
\end{aligned}$$

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

$$
\begin{aligned}
&\mathbb{E}[cost(\pi(\cdot|\mathcal{H}_{m-1}))] \\
&= c_{cs}F(L_0|\mathcal{H}_{m-1}) + \sum_{j=0}^{k-1} c_p\Big(Z(\mathcal{L},j-1) + L_{2j+1} - L_{2j}\Big)F(L_{2j+1}|\mathcal{H}_{m-1}) - \sum_{j=0}^{k-1} c_p Z(\mathcal{L},j-1)F(L_{2j}|\mathcal{H}_{m-1}) \\
&\quad - \sum_{j=0}^{k-1}\int_{L_{2j}}^{L_{2j+1}} c_p F(x|\mathcal{H}_{m-1})\,dx + \sum_{j=0}^{k-2}\Big(c_{cs} + c_p\cdot Z(\mathcal{L},j)\Big)\cdot F(L_{2j+2}|\mathcal{H}_{m-1}) \\
&\quad - \sum_{j=0}^{k-2}\Big(c_{cs} + c_p\cdot Z(\mathcal{L},j)\Big)F(L_{2j+1}|\mathcal{H}_{m-1}) + \Big(c_{cs} + c_p\cdot Z(\mathcal{L},k-1)\Big)\cdot 1 \\
&\quad - \Big(c_{cs} + c_p\cdot Z(\mathcal{L},k-1)\Big)F(L_{2k-1}|\mathcal{H}_{m-1}) \qquad (3) \\
&= c_{cs}F(L_0|\mathcal{H}_{m-1}) \; + \sum_{j=0}^{k-1} c_p Z(\mathcal{L},j)F(L_{2j+1}|\mathcal{H}_{m-1}) - \sum_{j=0}^{k-1} c_p Z(\mathcal{L},j-1)F(L_{2j}|\mathcal{H}_{m-1}) \\
&\quad - \sum_{j=0}^{k-1}\int_{L_{2j}}^{L_{2j+1}} c_p F(x|\mathcal{H}_{m-1})\,dx \quad + \sum_{j=0}^{k-2} c_{cs}\cdot F(L_{2j+2}|\mathcal{H}_{m-1}) + \sum_{j=0}^{k-2} c_p\cdot Z(\mathcal{L},j)\cdot F(L_{2j+2}|\mathcal{H}_{m-1}) \\
&\quad - \sum_{j=0}^{k-2} c_{cs}\cdot F(L_{2j+1}|\mathcal{H}_{m-1}) \quad - \sum_{j=0}^{k-2} c_p\cdot Z(\mathcal{L},j)\cdot F(L_{2j+1}|\mathcal{H}_{m-1}) \\
&\quad + c_{cs} + c_p\cdot Z(\mathcal{L},k-1) \; - c_{cs}\cdot F(L_{2k-1}|\mathcal{H}_{m-1}) \quad - c_p\cdot Z(\mathcal{L},k-1)\cdot F(L_{2k-1}|\mathcal{H}_m) \qquad (4)
\end{aligned}
$$

We apply integration by parts to the term $\int_{L_{2j}}^{L_{2j+1}} c_p\cdot\big(Z(\mathcal{L},j-1)+x-L_{2j}\big)\cdot f(x|\mathcal{H}_{m-1})\,dx$ in Equation (1) to get the terms $c_p\cdot\big(Z(\mathcal{L},j-1)+x-L_{2j}\big)\cdot F(x|\mathcal{H}_{m-1})\big|_{L_{2j}}^{L_{2j+1}} - \int_{L_{2j}}^{L_{2j+1}} c_p F(x|\mathcal{H}_{m-1})\,dx$ in Equation (2). In Equation (4), we have substituted $Z(\mathcal{L},j)$ for the terms $Z(\mathcal{L},j-1)+(L_{2j+1}-L_{2j})$ in Equation (3) since, $Z(\mathcal{L},j) = Z(\mathcal{L},j-1)+(L_{2j+1}-L_{2j})$. The remainder of the proof consists of combining and canceling terms to simplify (4), then applying the fundamental theorem of calculus.

$$
\begin{aligned}
&\mathbb{E}[cost(\pi(\cdot|\mathcal{H}_{m-1}))] \\
&= \; c_{cs}F(L_0|\mathcal{H}_{m-1}) \; + \sum_{j=0}^{k-1} c_p Z(\mathcal{L},j)\cdot F(L_{2j+1}|\mathcal{H}_{m-1}) - \sum_{j=0}^{k-1} c_p Z(\mathcal{L},j-1)\cdot F(L_{2j}|\mathcal{H}_{m-1}) \\
&\quad - \sum_{j=0}^{k-1}\int_{L_{2j}}^{L_{2j+1}} c_p F(x|\mathcal{H}_{m-1})\,dx \quad + \sum_{j=0}^{k-2} c_{cs}\cdot F(L_{2j+2}|\mathcal{H}_{m-1}) \quad + \sum_{j=0}^{k-2} c_p\cdot Z(\mathcal{L},j)\cdot F(L_{2j+2}|\mathcal{H}_{m-1}) \\
&\quad - \sum_{j=0}^{k-1} c_{cs}\cdot F(L_{2j+1}|\mathcal{H}_{m-1}) \quad - \sum_{j=0}^{k-1} c_p\cdot Z(\mathcal{L},j)\cdot F(L_{2j+1}|\mathcal{H}_{m-1}) + c_{cs} + c_p\cdot Z(\mathcal{L},k-1) \\
&= \; -\sum_{j=0}^{k-1}\int_{L_{2j}}^{L_{2j+1}} c_p F(x|\mathcal{H}_{m-1})\,dx + \sum_{j=0}^{k-1} c_{cs}F(L_{2j}|\mathcal{H}_{m-1}) - \sum_{j=0}^{k-1} c_{cs}F(L_{2j+1}|\mathcal{H}_{m-1}) + c_{cs} + c_p Z(\mathcal{L},k-1) \qquad (5) \\
&= \quad c_p\cdot Z(\mathcal{L},k-1) - \sum_{j=0}^{k-1}\int_{L_{2j}}^{L_{2j+1}} c_p F(x|\mathcal{H}_{m-1})\,dx - \sum_{j=0}^{k-1} c_{cs}\cdot\Big(F(L_{2j+1}|\mathcal{H}_{m-1}) - F(L_{2j}|\mathcal{H}_{m-1})\Big) + c_{cs} \\
&= \quad \int_{\mathcal{I}} c_p(1-F(x|\mathcal{H}_{m-1})) - c_{cs}\cdot f(x|\mathcal{H}_{m-1})\,dx \; + \; c_{cs}
\end{aligned}
$$

where $\mathcal{I} = \begin{cases} 1, \text{ for } \quad x\in[L_0,\; L_1]\bigcup\cdots\bigcup[L_{2k-2}, L_{2k-1}] \\ 0, \text{ otherwise} \end{cases}$.

In Equation (5), we combine $c_{cs} \cdot F(L_0|\mathcal{H}_{m-1})$ and $\sum_{j=0}^{k-2} c_{cs}F(L_{2j+2}|\mathcal{H}_{m-1})$ to obtain $\sum_{j=0}^{k-1} c_{cs}F(L_{2j}|\mathcal{H}_{m-1})$.

□

Theorem 2. *The points L_i of the sequence of keep-alive windows over an inter-arrival for the optimal policy $\pi_{opt}(\cdot|\mathcal{H}_{m-1})$ are at 0, ∞, or solutions to the equation $c_p - (c_{cs} \cdot \lambda(x|\mathcal{H}_{m-1})) = 0$ where the sign changes.*

Proof. From Lemma 1 we know that the expected cost is given by $\mathbb{E}[cost(\pi(\cdot|\mathcal{H}_{m-1}))] = \int_0^\infty \pi(x|\mathcal{H}_{m-1}) \cdot g(x|\mathcal{H}_{m-1})\ dx + c_{cs}$. The points of the sequence of keep-alive windows L_k for $k = 0, 1, 2, \cdots$ for the optimal policy are the points where the first order partial derivative of the expected cost is zero, that is, $\frac{\partial\, \mathbb{E}[cost(\pi(\cdot|\mathcal{H}_{m-1}))]}{\partial x} = 0$ at $x = L_k\ \forall k$. The first order derivative of the expected cost is $\frac{\partial\, \mathbb{E}[cost(\pi(\cdot|\mathcal{H}_{m-1}))]}{\partial x} = g(x|\mathcal{H}_{m-1})$. We simplify $g(x|\mathcal{H}_{m-1})$ as follows,

$$
\begin{aligned}
&g(x|\mathcal{H}_{m-1}) \\
&= c_p(1 - F(x|\mathcal{H}_{m-1})) - c_{cs}f(x|\mathcal{H}_{m-1}) \\
&= c_p(1 - F(x|\mathcal{H}_{m-1})) - c_{cs}\lambda(x|\mathcal{H}_{m-1})(1 - F(x|\mathcal{H}_{m-1})) \\
&= (1 - F(x|\mathcal{H}_{m-1})) \cdot (c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1}))
\end{aligned}
$$

$$
\begin{aligned}
& g(x = L_k|\mathcal{H}_{m-1}) = 0 \\
\implies & 1 - F(x = L_k|\mathcal{H}_{m-1}) = 0 \quad \text{or} \quad c_p - c_{cs}\lambda(x = L_k|\mathcal{H}_{m-1}) = 0 \\
\implies & F(x = L_k|\mathcal{H}_{m-1}) = 1 \quad \text{or} \quad \frac{c_p}{c_{cs}} = \lambda(x = L_k|\mathcal{H}_{m-1}) = \frac{f(x = L_k|\mathcal{H}_{m-1})}{1 - F(x = L_k|\mathcal{H}_{m-1})}
\end{aligned}
$$

Hence, the points where $g(x|\mathcal{H}_{m-1}) = 0$ are also the points where $c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1}) = 0$. We know that the instantaneous cost of the policy over an inter-arrival is given by $g(x|\mathcal{H}_{m-1})$. Let $x = L_k$ be an arbitrary root of the equation $c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1}) = 0$. If $g(x|\mathcal{H}_{m-1})$ changes sign from positive to negative as it goes through $x = L_k$, that is, $g(x|\mathcal{H}_{m-1}) > 0$ for $x < L_k$ and changes sign to $g(x|\mathcal{H}_{m-1}) < 0$ for $x > L_k$. It would be optimal for the cache policy to start the keep-alive window from $x = L_k$, since the cost of caching the object from $x = L_k$ benefits the policy. Similarly, if $g(x|\mathcal{H}_{m-1})$ changes sign from negative to positive as it passes $x = L_k$, that is, $g(x|\mathcal{H}_{m-1}) < 0$ for $x < L_k$ and changes sign to $g(x|\mathcal{H}_{m-1}) > 0$ for $x > L_k$. It would be optimal to stop the keep-alive window after $x = L_k$, since the cost of caching the object after $x = L_k$ will not benefit the policy. Since $1 - F(x|\mathcal{H}_{m-1}) \geq 0, \quad \forall x,$ the sign of $c_p - c_{cs} \cdot \lambda(x|\mathcal{H}_{m-1})$ determines the sign of $g(x|\mathcal{H}_{m-1})$. The sign of $c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1})$ should change as it passes through the root of the equation $x = L_k$ for $x = L_k$ to be considered as a point where the keep-alive window of the optimal policy starts or ends.

This leaves the end cases where there is no solution to the equation $c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1}) = 0$ or the sign of $c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1})$ does not change $\forall\ x$. When $g(x|\mathcal{H}_{m-1})$ is always non-negative, it is optimal to have a keep-alive window length of 0. This is because having an active keep-alive window length in this case would be more expensive than a cold start. On the other hand, when $g(x|\mathcal{H}_{m-1})$ is always non-positive, it is optimal for the keep-alive window to always be active since keeping the object in cache is beneficial to the policy.

□

Corollary 2.1. *If the hazard rate is weakly decreasing, the optimal policy $\pi_{opt}(\cdot|\mathcal{H}_{m-1})$ is a single keep-alive window starting at $\tau_{pw} = 0$, and is given by*

$$\pi_{opt}(x|\mathcal{H}_{m-1}) = \begin{cases} 1\ , & \forall x \in [0,\ \tau_{opt,\mathcal{H}_{m-1}}] \\ 0\ , & \textit{otherwise} \end{cases} \quad \textit{where,}$$

1. *$\tau_{opt,\mathcal{H}_{m-1}} = \infty$, i.e., the optimal policy is to have the keep-alive window always be active when $\forall x,\ \frac{c_p}{c_{cs}} < \lambda(x|\mathcal{H}_{m-1})$,*

2. $\tau_{opt,\mathcal{H}_{m-1}} = 0$, *i.e., the optimal policy would be to not cache and always have a cold start when* $\frac{c_p}{c_{cs}} > \lambda(x = 0|\mathcal{H}_{m-1})$
3. *The optimal policy is a keep-alive window of length* $\tau_{opt,\mathcal{H}_{m-1}}$ *given by the solution to the equation* $\frac{c_p}{c_{cs}} = \frac{f(x = \tau_{opt,\mathcal{H}_{m-1}}|\mathcal{H}_{m-1})}{1 - F(x = \tau_{opt,\mathcal{H}_{m-1}}|\mathcal{H}_{m-1})}$, *otherwise.*

Proof. Assume that the hazard rate of the arrival of function invocations over an inter-arrival $\lambda(x|\mathcal{H}_{m-1})$ is (weakly) decreasing. From Lemma 1 we know that the expected cost is given by $\mathbb{E}[cost(\pi(\cdot|\mathcal{H}_{m-1}))] = \int_0^\infty \pi(x|\mathcal{H}_{m-1}) \cdot g(x|\mathcal{H}_{m-1})\ dx + c_{cs}$. We prove a single keep-alive window is optimal by showing that $g(x|\mathcal{H}_{m-1})$ can only change its sign from negative to positive at most once. Thus g is optimized by single window policy that keeps the object in cache until the transition from negative to positive occurs. To begin,

$$g(x|\mathcal{H}_{m-1}) = \ (1 - F(x|\mathcal{H}_{m-1})) \cdot (c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1}))$$

Since $\lambda(x|\mathcal{H}_{m-1})$ is weakly decreasing, $c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1})$ is weakly increasing. Also, $1 - F(x|\mathcal{H}_{m-1})$ is always positive. If $g(x|\mathcal{H}_{m-1}) \geq 0$, then the optimal policy is to have a keep-alive window length of 0. This is because having an active keep-alive window in this case would be more expensive than a cold start. If $g(x|\mathcal{H}_{m-1})$ is always negative, then it is always beneficial for the keep-alive window to be active. Otherwise, $g(x|\mathcal{H}_{m-1})$ can change its sign at most once and such a change must be from negative to positive. It is no longer beneficial for the provider to keep things in memory after $g(x|\mathcal{H}_{m-1})$ has changed from negative to positive because the cost of keeping in memory outweighs the cost of a cold start. Thus, the optimal policy is of the form of a single keep-alive window. Now, it only remains to determine the point$\tau_{\text{opt},\mathcal{H}_{m-1}}$ at which $g(x|\mathcal{H}_{m-1})$ changes from negative to positive.

$$\begin{aligned} & g(x = \tau_{\text{opt},\mathcal{H}_{m-1}}|\mathcal{H}_{m-1}) \ = \ 0 \\ \implies & F(x = \tau_{\text{opt},\mathcal{H}_{m-1}}|\mathcal{H}_{m-1}) = 1 \quad \text{or} \quad \frac{c_p}{c_{cs}} = \lambda(x = \tau_{\text{opt},\mathcal{H}_{m-1}}|\mathcal{H}_{m-1}) = \frac{f(x = \tau_{\text{opt},\mathcal{H}_{m-1}}|\mathcal{H}_{m-1})}{1 - F(x = \tau_{\text{opt},\mathcal{H}_{m-1}}|\mathcal{H}_{m-1})} \end{aligned}$$

□

Corollary 2.2. *If the hazard rate is weakly increasing, the optimal policy* $\pi_{opt}(\cdot|\mathcal{H}_{m-1})$ *is a single keep-alive window with* $\tau_{ka} = \infty$ *and a pre-warming window, and is given by*

$$\pi_{opt}(x|\mathcal{H}_{m-1}) = \begin{cases} 1, \ \tau_{pw,\mathcal{H}_{m-1}} \leq x \\ 0, \ \textit{otherwise} \end{cases} \quad \textit{where,}$$

1. $\tau_{pw,\mathcal{H}_{m-1}} = 0$, *i.e., the optimal policy is to have the keep-alive window always be active when* $\forall x, \quad \frac{c_p}{c_{cs}} < \lambda(x|\mathcal{H}_{m-1})$,
2. $\tau_{pw,\mathcal{H}_{m-1}} = \infty$, *i.e., the optimal policy is to always have a cold start when* $\forall x, \ \frac{c_p}{c_{cs}} > \lambda(x|\mathcal{H}_{m-1})$.
3. $\tau_{pw,\mathcal{H}_{m-1}}$ *satisfies the equation* $\frac{c_p}{c_{cs}} = \frac{f(x = \tau_{pw,\mathcal{H}_{m-1}}|\mathcal{H}_{m-1})}{1 - F(x = \tau_{pw,\mathcal{H}_{m-1}}|\mathcal{H}_{m-1})}$, *i.e., an infinite keep-alive window after a pre-warming window of length* $\tau_{pw,\mathcal{H}_{m-1}}$ *when* $c_p - c_{cs}\lambda(x = 0|\mathcal{H}_{m-1}) > 0$ *and changes sign.*

Proof. Following the proof of Theorem 2, we know that $g(x|\mathcal{H}_{m-1}) = (1 - F(x|\mathcal{H}_{m-1})) \cdot (c_p - c_{cs} \cdot \lambda(x|\mathcal{H}_{m-1}))$. Since $\lambda(x|\mathcal{H}_{m-1})$ is weakly increasing, $c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1})$ is weakly decreasing. Also, $1 - F(x|\mathcal{H}_{m-1})$ is always positive. If initially $g(x|\mathcal{H}_{m-1}) < 0$, then $g(x|\mathcal{H}_{m-1})$ will always be negative. Hence, it is optimal to have the keep-alive window always be active. If $g(x|\mathcal{H}_{m-1}) > 0, \quad \forall x$, that is, g is always positive, then the optimal policy is to encounter a cold start. If $g(x|\mathcal{H}_{m-1})$ is initially positive, then changes to a negative sign as $\lambda(x|\mathcal{H}_{m-1})$ is weakly increasing, then the optimal policy will be a pre-warming window of length decided by the position of the change of sign. We obtain $\tau_{\text{pw},\mathcal{H}_{m-1}}$ from solving $g(x = \tau_{\text{pw},\mathcal{H}_{m-1}}|\mathcal{H}_{m-1}) = 0$ as follows.

$$
\begin{aligned}
& g(x = \tau_{\text{pw},\mathcal{H}_{m-1}} | \mathcal{H}_{m-1}) \;=\; 0 \\
\implies & F(x = \tau_{\text{pw},\mathcal{H}_{m-1}} | \mathcal{H}_{m-1}) = 1 \quad \text{or} \quad \frac{c_p}{c_{cs}} = \lambda(x = \tau_{\text{pw},\mathcal{H}_{m-1}} | \mathcal{H}_{m-1}) = \frac{f(x = \tau_{\text{pw},\mathcal{H}_{m-1}} | \mathcal{H}_{m-1})}{1 - F(x = \tau_{\text{pw},\mathcal{H}_{m-1}} | \mathcal{H}_{m-1})}
\end{aligned}
$$

After the sign changes to negative, the keep-alive window should always be active,, that is, $\tau_{\text{ka},\mathcal{H}_{m-1}} = \infty$. □

Corollary 2.1 characterizes the optimal policy when the distribution of arrival requests follow the Hawkes process to be one of the following policies,

- The keep-alive window is to always be active with $\tau_{\text{opt},\mathcal{H}_{m-1}} = \infty$ when, as in the Poisson case, the background intensity is sufficiently high: $\frac{c_p}{c_{cs}} < \lambda_0$.
- Experience a cold start with $\tau_{\text{opt},\mathcal{H}_{m-1}} = 0$ when $\frac{c_p}{c_{cs}} > \lambda(x|\mathcal{H}_{m-1}))$, after the most recent arrival request.
- The keep-alive window is given by the expression

$$
\tau_{\text{opt},\mathcal{H}_{m-1}} = \frac{1}{\beta}\Big(\log \alpha + \log \Big(\sum_{j=1}^{m-1} e^{\beta(t_j - t_{m-1})} \Big) - \log \Big(\frac{c_p}{c_{cs}} - \lambda_0 \Big) \Big)
$$

otherwise.
To compute $\tau_{\text{opt},\mathcal{H}_{m-1}}$, we know that the length of the optimal keep-alive window is $\tau_{\text{opt},\mathcal{H}} = t_{\text{opt},\mathcal{H}} - \; t_{m-1}$, where t_{m-1} is the most recent arrival request. This expression is obtained by substituting the conditional intensity of the Hawkes process in Corollary 2.1 and solving for $t_{\text{opt},\mathcal{H}_{m-1}}$.

$$
\begin{aligned}
\frac{c_p}{c_{cs}} &= \lambda_0 + \sum_{t_j \in \mathcal{H}_{m-1}} \alpha \cdot e^{-\beta \cdot (t_{\text{opt},\mathcal{H}_{m-1}} - t_j)} \\
\frac{1}{\alpha}\Big(\frac{c_p}{c_{cs}} - \lambda_0 \Big) &= \sum_{t_j \in \mathcal{H}_{m-1}} e^{-\beta \cdot (t_{\text{opt},\mathcal{H}_{m-1}} - t_j)} \\
\frac{1}{\alpha}\big(\frac{c_p}{c_{cs}} - \lambda_0 \Big) &= e^{-\beta \cdot t_{\text{opt},\mathcal{H}_{m-1}}} \cdot \sum_{t_j \in \mathcal{H}_{m-1}} e^{\beta \cdot t_j} \\
\log \left(\frac{1}{\alpha}\big(\frac{c_p}{c_{cs}} - \lambda_0 \Big) \right) &= -\beta \cdot t_{\text{opt},\mathcal{H}_{m-1}} + \log \left(\sum_{t_j \in \mathcal{H}_{m-1}} e^{\beta \cdot t_j} \right) \\
\beta \cdot t_{\text{opt},\mathcal{H}_{m-1}} &= \log \alpha + \log \left(\sum_{t_j \in \mathcal{H}_{m-1}} e^{\beta \cdot t_j} \right) - \log \Big(\frac{c_p}{c_{cs}} - \lambda_0 \Big) \\
\beta \cdot t_{\text{opt},\mathcal{H}_{m-1}} &= \log \alpha + \log \left(\Big(e^{\beta \cdot t_{m-1}} \Big) \cdot \Big(\sum_{t_j \in \mathcal{H}_{m-1}} e^{\beta \cdot t_j - \beta \cdot t_{m-1}} \Big) \right) - \log \Big(\frac{c_p}{c_{cs}} - \lambda_0 \Big) \\
\beta \cdot t_{\text{opt},\mathcal{H}_{m-1}} &= \log \alpha + \beta \cdot t_{m-1} + \log \left(\sum_{t_j \in \mathcal{H}_{m-1}} e^{\beta \cdot (t_j - t_{m-1})} \right) - \log \Big(\frac{c_p}{c_{cs}} - \lambda_0 \Big) \\
t_{\text{opt},\mathcal{H}_{m-1}} &= t_{m-1} + \frac{1}{\beta} \left(\log \alpha + \log \left(\sum_{t_j \in \mathcal{H}_{m-1}} e^{\beta \cdot (t_j - t_{m-1})} \right) - \log \Big(\frac{c_p}{c_{cs}} - \lambda_0 \Big) \right) \\
\tau_{\text{opt},\mathcal{H}_{m-1}} &= \frac{1}{\beta} \cdot \left(\log \alpha + \log \big(\sum_{t_j \in \mathcal{H}_{m-1}} e^{\beta \cdot (t_j - t_{m-1})} \big) - \log \Big(\frac{c_p}{c_{cs}} - \lambda_0 \Big) \right)
\end{aligned}
$$

Corollary 2.3. *When the parameters of the Hawkes process are such that $c_p - (c_{cs} \cdot \lambda(x|\mathcal{H})) = 0$ has a solution, the optimal policy has a history independent lower bound, and an upper bound expressed as follows*

$$\tau_{opt,\mathcal{H}} \geq \frac{1}{\beta} \cdot \left(\log \alpha - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right)$$
$$\tau_{opt,\mathcal{H}} \leq \frac{1}{\beta} \cdot \left(\log \alpha + \log \delta + 1 - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right)$$

where δ satisfies

$$\sum_{i=m-\delta}^{m-1} e^{\beta \cdot (t_i - t_{m-1})} \geq \frac{1}{2} \sum_{i=1}^{m-1} e^{\beta \cdot (t_i - t_{m-1})}$$

Proof. We can rewrite the formula for the optimal policy for a Hawkes process with a given history as:

$$\begin{aligned}
\tau_{\text{opt},\mathcal{H}} &= t_{\text{opt},\mathcal{H}} - t_m \\
&= \frac{1}{\beta} \cdot \left(\log \alpha + \log \left(\sum_{t_j \in \mathcal{H}} e^{\beta \cdot t_j} \right) - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right) - t_m \\
&= \frac{1}{\beta} \cdot \left(\log \alpha + \log \left(e^{\beta \cdot t_1} + e^{\beta \cdot t_2} + \cdots + e^{\beta \cdot t_m} \right) - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right) - t_m \\
&= \frac{1}{\beta} \cdot \left(\log \alpha + \log \left(e^{\beta \cdot t_1} + e^{\beta \cdot t_2} + \cdots + e^{\beta \cdot t_m} \right) - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right) - t_m \\
&= \frac{1}{\beta} \cdot \left(\log \alpha + \log \left((e^{\beta \cdot t_m}) \cdot (e^{\beta \cdot (t_1 - t_m)} + e^{\beta \cdot (t_2 - t_m)} + \cdots + e^{\beta \cdot (t_m - t_m)}) \right) - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right) - t_m \\
&= \frac{1}{\beta} \cdot \left(\log \alpha + \log \left(e^{\beta \cdot (t_1 - t_m)} + e^{\beta \cdot (t_2 - t_m)} + \cdots + e^{\beta \cdot (t_m - t_m)} \right) - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right) \\
&= \frac{1}{\beta} \cdot \left(\log \alpha + \log \left(e^{\beta \cdot (t_1 - t_m)} + e^{\beta \cdot (t_2 - t_m)} + \cdots + 1 \right) - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right)
\end{aligned}$$

This has three terms, two of which are independent of the history. Thus we can obtain a lower bound on the optimal policy for *any* history as$\tau_{\text{opt},\mathcal{H}} \geq \frac{1}{\beta} \cdot \left(\log \alpha - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right)$. In fact, this is the optimal policy for the empty history.

For the term that depends on history, all exponents are negative so each term is at most 1. This yields a trivial upper bound of $\tau_{\text{opt},\mathcal{H}} \leq \frac{1}{\beta} \cdot \left(\log \alpha + \log m - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right)$. While it grows slowly due to the log, this bound is unappealing to apply directly because it grows with the length of the history. In reality, many of the terms of the sum are close to 0 because $t_i - t_m$ is very negative for t_i substantially in the past.

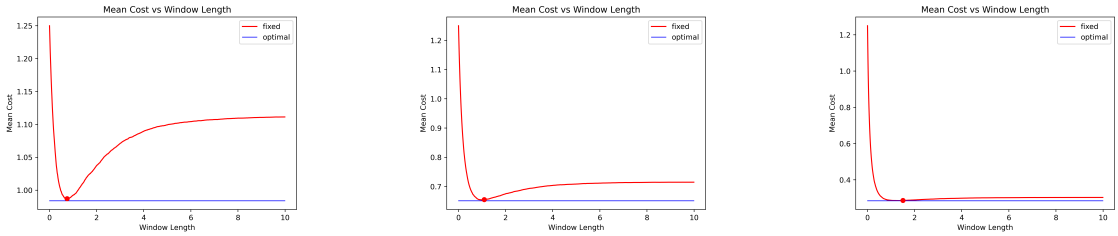
To get a better estimate, let δ be such that

$$\sum_{i=m-\delta+1}^{m} e^{\beta \cdot (t_i - t_m)} \geq \frac{1}{2} \sum_{i=1}^{m} e^{\beta \cdot (t_i - t_m)}$$

That is, the most recent δ arrivals provide at least half the total weight. This can be thought of as only having δ arrivals that are recent enough to matter. Then we have the upper bound of $\tau_{\text{opt},\mathcal{H}} \leq \frac{1}{\beta} \cdot \left(\log \alpha + \log \delta + 1 - \log \left(\frac{c_p}{c_{cs}} - \lambda_0 \right) \right)$. □

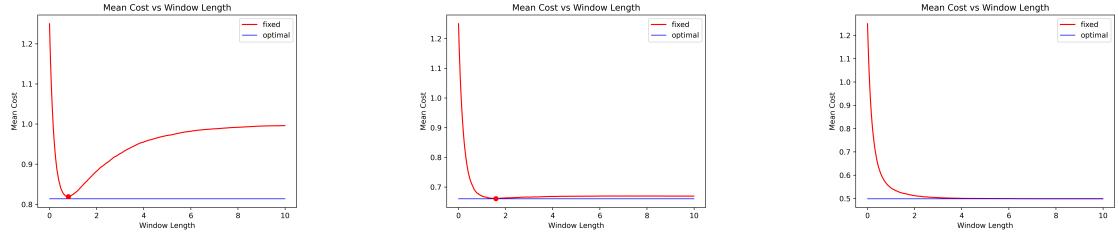
B OMITTED FIGURES FROM SECTION 5.1

These additional figures demonstrate the robustness of the performance of Optimized-TTL with respect to a range of parameters. In them, we examine how the average costs behave with respect to the Hawkes process parameters λ_0, α, and β



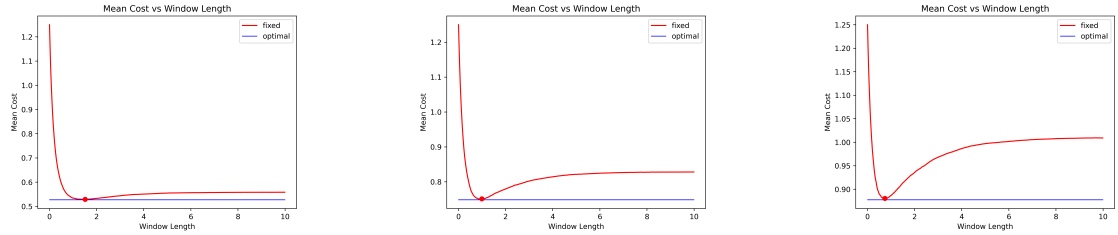
(a) $\alpha = 0.8$ (b) $\alpha = 1.4$ (c) $\alpha = 2.0$

Figure 1: Plots of average costs for policies when α is increased, given $\lambda_0 = 0.6, \beta = 2.4, c_p = 1.0, c_{cs} = 1.25$



(a) $\lambda_0 = 0.5$ (b) $\lambda_0 = 0.75$ (c) $\lambda_0 = 1.0$

Figure 2: Plots of average costs for policies when λ_0 is increased, given $\alpha = 1.2, \beta = 2.4, c_p = 1.0, c_{cs} = 1.25$



(a) $\beta = 1.8$ (b) $\beta = 2.4$ (c) $\beta = 3.0$

Figure 3: Plots of average costs for policies when β is increased, given $\lambda_0 = 0.6, \alpha = 1.2, c_p = 1.0, c_{cs} = 1.25$

while holding the costs fixed. Figure 1 shows the behavior of the average cost of the policies for different values of α of the Hawkes process. We see that as α increases, the average length of the optimal keep-alive window increases. This is because for higher values of α, the intensity of the subsequent arrival will be larger making it larger keep-alive windows more desirable. This intuition can be made more precise with Corollary 2.1 since $g(x|\mathcal{H}_{m-1}) = (1 - F(x|\mathcal{H}_{m-1})) \cdot (c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1}))$. Increasing α, increases $\lambda(x|\mathcal{H}_{m-1})$ causing $g(x|\mathcal{H}_{m-1})$ to be more negative. Therefore, the point τ_{opt} where $g(x|\mathcal{H}_{m-1})$ changes sign from negative to positive is larger for a larger α. The behavior of the average costs of the policies when λ_0 increases is similar to that of α as shown in Figure 2. From Figure 3 we see that as β increases, the average length of the optimal keep-alive window decreases. The decay rate of the arrivals' influence is larger for a larger β which makes shorter keep-alive windows more optimal. This connects to Corollary 2.1, where a higher value of β causes $g(x|\mathcal{H}_{m-1}) = (1 - F(x|\mathcal{H}_{m-1})) \cdot (c_p - c_{cs}\lambda(x|\mathcal{H}_{m-1}))$ to change from negative to positive earlier.

C EXTENSION: WORST-CASE GUARANTEES FOR HAWKES PROCESSES

We know from Theorem 2, that computing the optimal keep-alive policy requires the history $\mathcal{H}_{m-1}$ of previous $m-1$ invocations. As described in Section 4.2, the computational complexity of the optimal policy increases as the history of invocations increase. Hence, we propose history independent policies that do not require any information regarding past arrival requests. This problem is similar in spirit to the *Ski Rental* problem in online algorithms, where the customer can buy an item for $ B or rent the item for $ R per the unit of time. There, a 2-approximation results from renting until

the cost of buying has been paid in rental fees as if the input ends during the rental period the policy was optimal and otherwise buying immediately would have been optimal so the policy overpaid by a factor of 2. Similarly, in our setting a fixed keep-alive policy can achieve a 2-approximation (Theorem 4). This bound does not use any information about the parameters of the Hawkes process. In Theorem 5, we propose a history independent approximate policy that requires only the parameters of the Hawkes process (i.e. is independent of the history), and approximates the optimal cost by a factor of $1 + \left(\frac{1}{\frac{c_p}{c_{cs}} \cdot \tau_{\text{opt},\mathcal{H}=\phi} + 1} \right)^{\frac{1}{2}}$. Both results follow from the following lemma, which bounds the performance of arbitrary history independent keep-alive policies.

Lemma 3. *A policy with keep-alive window τ which does not depend on the history of arrivals of invocations is at least* $\max\left\{ \frac{c_p \cdot \tau}{c_p \cdot \tau_{opt,\mathcal{H}=\phi} + c_{cs}} + 1, 1 + \frac{c_{cs}}{c_p \cdot \tau} \right\}$ *approximation to the cost of the optimal policy $\tau_{opt,\mathcal{H}_{m-1}}$ for any history $\mathcal{H}_{m-1}$.*

Proof. Given history of application invocations $\mathcal{H}_{m-1}$, we denote the length of the optimal keep-alive window by $\tau_{\text{opt},\mathcal{H}_{m-1}}$. Let τ denote the length of a history independent policy. We examine the upper bound of the ratio of the cost of the history independent policy to the cost of the optimal policy when the application is invoked at the m-th inter-arrival x_m, that is, $\frac{cost(x_m, \tau)}{cost(x_m, \tau_{\text{opt},\mathcal{H}_{m-1}})}$, where $x_m = t_m - t_{m-1}$ is the length of the m-th inter-arrival. There are three possibilities when comparing keep-alive window τ with $\tau_{\text{opt},\mathcal{H}_{m-1}}$. They are,

1. $\tau = \tau_{\text{opt},\mathcal{H}_{m-1}}$
2. $\tau < \tau_{\text{opt},\mathcal{H}_{m-1}}$
3. $\tau > \tau_{\text{opt},\mathcal{H}_{m-1}}$

We examine the upper bound of the ratio of the cost of the history independent policy to the cost of the optimal policy for each case listed above.

Case 1: When $\tau = \tau_{\text{opt},\mathcal{H}_{m-1}}$, both the history independent policy and the optimal policy have the same cost. That is ,

$$cost(x_m, \tau) = cost(x_m, \tau_{\text{opt},\mathcal{H}_{m-1}})$$

Case 2: When $\tau < \tau_{\text{opt},\mathcal{H}_{m-1}}$, the cost of the policies can be compared based on when the application invocation occurs.

- When the application invocation occurs before the end of the history independent keep-alive window, that is, $x_m \leq \tau$, then both policies encounter a warm start. Hence, both policies have the same cost. That is,

$$cost(x_m, \tau) = cost(x_m, \tau_{\text{opt},\mathcal{H}_{m-1}}) = c_p \cdot x_m$$

- When the application invocation is after the keep-alive window τ, but before the end of the optimal policy, that is, $\tau < x_m \leq \tau_{\text{opt},\mathcal{H}_{m-1}}$, then the history independent policy encounters a cold start whereas the optimal policy experiences a warm start. The ratio of the cost of the history independent policy to the cost of the optimal policy is expressed as follows,

$$\begin{aligned} \frac{cost(x_m, \tau)}{cost(x_m, \tau_{\text{opt},\mathcal{H}_{m-1}})} &= \frac{c_p \cdot \tau + c_{cs}}{c_p \cdot x_m} \\ &\leq \frac{c_p \cdot \tau + c_{cs}}{c_p \cdot \tau} \\ &= 1 + \frac{c_{cs}}{c_p \cdot \tau} \end{aligned} \tag{6}$$

  In Equation (6) above we see that the minimum possible cost of the optimal policy in this scenario is when the application gets invoked just after the fixed keep-alive window, that is, when $x_m = \tau$.

- When the application invocation is after the optimal keep-alive policy, that is, $\tau < \tau_{\text{opt},\mathcal{H}_{m-1}} < x_m$, then both policies encounter a cold start. Here, the cost of the optimal policy is larger than the cost of the history independent

keep-alive policy because the cost of a policy when a cold start occurs is proportional to the length of the keep-alive window. That is,

$$\begin{aligned} \tau &\leq \tau_{\text{opt},\mathcal{H}_{m-1}} \\ \implies c_p \cdot \tau + c_{cs} &\leq c_p \cdot \tau_{\text{opt},\mathcal{H}_{m-1}} + c_{cs} \\ \implies cost(x_m, \tau) &\leq cost(x_m, \tau_{\text{opt},\mathcal{H}_{m-1}}) \end{aligned}$$

Case 3: When $\tau > \tau_{\text{opt},\mathcal{H}_{m-1}}$, the cost of the policies can be compared based on the arrival of application invocations.

- When the application is invoked before the end of the optimal keep-alive window, that is, $x_m \leq \tau_{\text{opt},\mathcal{H}_{m-1}}$, then both the policies encounter a warm start. Hence, both the policies have the same costs. That is,

$$cost(x_m, \tau) = cost(x_m, \tau_{\text{opt},\mathcal{H}_{m-1}}) = c_p \cdot x_m$$

- When the application invocation occurs after the optimal keep-alive window, that is, $x_m > \tau_{\text{opt},\mathcal{H}_{m-1}}$, then the optimal policy experiences a cold start. The ratio of the cost of the history independent policy to the cost of the optimal policy is upper bounded when the history independent policy has a cold start. We compute the upper bound on the ratio of costs as follows,

$$\begin{aligned} \frac{cost(x_m, \tau)}{cost(x_m, \tau_{\text{opt},\mathcal{H}_{m-1}})} &\leq \frac{c_p \cdot \tau + c_{cs}}{c_p \cdot \tau_{\text{opt},\mathcal{H}_{m-1}} + c_{cs}} \\ &\leq \frac{c_p \cdot \tau}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} + 1 \qquad (7) \end{aligned}$$

where in Equation (7) we have substituted $\mathcal{H} = \phi$ to compute the upper bound.

We have now established two separate upper bounds for cases 2 and 3. The approximation factor of the history independent policy with respect to the optimal policy for an arbitrary history is the maximum of the two upper bounds, that is, $\max\left\{\frac{c_p \cdot \tau}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} + 1, 1 + \frac{c_{cs}}{c_p \cdot \tau}\right\}$.

□

C.1 FIXED KEEP-ALIVE POLICY

We first show our Ski rental style result.

Theorem 4. *The cost of the fixed keep-alive policy $\tau_{fixed} = c_{cs}/c_p$ is at most twice the cost of the optimal keep-alive policy $\tau_{opt,\mathcal{H}_{m-1}}$. That is, when a function is invoked at time t_m after previous $m-1$ arrivals we have, $cost(x_m, \tau_{fixed}) \leq 2 \cdot cost(x_m, \tau_{opt,\mathcal{H}_{m-1}})$, where $x_m = t_m - t_{m-1}$ is the length of the m-th inter-arrival, and $c_p \cdot \tau_{fixed} = c_{cs}$.*

While this has a simple direct proof, we illustrate how it follows from Lemma 3.

Proof. From Lemma 3, we know that the approximation factor of the fixed policy with respect to the optimal policy is $\max\left\{\frac{c_p \cdot \tau_{\text{fixed}}}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} + 1, 1 + \frac{c_{cs}}{c_p \cdot \tau_{\text{fixed}}}\right\}$. The upper bound of $\frac{c_p \cdot \tau_{\text{fixed}}}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} + 1$ can further be reduced to $\frac{c_p \cdot \tau_{\text{fixed}}}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} + 1 \leq \frac{c_p \cdot \tau_{\text{fixed}}}{c_{cs}} + 1$ by substituting $\tau_{\text{opt},\mathcal{H}=\phi} = 0$ because a fixed policy should accommodate for any history independent policy.

The best length of the keep-alive window for the fixed policy is the length which minimizes the maximum of the two upper bounds on the ratio of the cost of the fixed policy and the optimal policy. Mathematically, the best length of the fixed policy is expressed as

$$\arg\min_{\tau_{\text{fixed}}} \max\left\{\frac{c_p \cdot \tau_{\text{fixed}}}{c_{cs}} + 1, 1 + \frac{c_{cs}}{c_p \cdot \tau_{\text{fixed}}}\right\}$$

We obtain the length of the fixed policy by solving the above expression,

$$
\begin{aligned}
\frac{c_p \cdot \tau_{\text{fixed}}}{c_{cs}} + 1 &= 1 + \frac{c_{cs}}{c_p \cdot \tau_{\text{fixed}}} \\
\frac{c_p \cdot \tau_{\text{fixed}}}{c_{cs}} &= \frac{c_{cs}}{c_p \cdot \tau_{\text{fixed}}} \\
\left(c_p \cdot \tau_{\text{fixed}}\right)^2 &= (c_{cs})^2 \\
c_p \cdot \tau_{\text{fixed}} &= c_{cs}
\end{aligned}
$$

Substituting this back to the upper bound on the ratio of the cost of the fixed policy to the cost of the optimal policy, we get

$$
\begin{aligned}
\frac{cost(x_m, \tau_{\text{fixed}})}{cost(x_m, \tau_{\text{opt},\mathcal{H}_{m-1}})} &\leq 1 + \frac{c_p \cdot \tau_{\text{fixed}}}{c_{cs}} \\
&= 1 + \frac{c_{cs}}{c_{cs}} \\
&= 2
\end{aligned}
$$

□

C.2 HISTORY INDEPENDENT KEEP-ALIVE POLICIES

More generally, we can take advantage of Lemma 3 to achieve a tighter bound that makes use of the parameters of the Hawkes process only through the policy they induce given the empty history.

Theorem 5. *There exists a policy with keep-alive window τ_{approx} which does not require the history of arrivals of application invocations with its cost upper bounded by a factor of $1 + \left(\frac{1}{\frac{c_p}{c_{cs}} \cdot \tau_{opt,\mathcal{H}=\phi}+1}\right)^{\frac{1}{2}}$ with respect to the cost of the optimal keep-alive policy $\tau_{opt,\mathcal{H}_{m-1}}$. In other words, for a given history of invocations $\mathcal{H}_{m-1}$, when the application invocation has an inter-arrival of length x_m, $\frac{cost(x_m, \tau_{approx})}{cost(x_m, \tau_{opt,\mathcal{H}_{m-1}})} \leq 1 + \left(\frac{1}{\frac{c_p}{c_{cs}} \cdot \tau_{opt,\mathcal{H}_{m-1}=\phi} + 1}\right)^{\frac{1}{2}} \leq 2$.*

Proof. From Lemma 3, we know that the approximation factor of the approximate policy with respect to the optimal policy is $\max\left\{\frac{c_p \cdot \tau_{\text{approx}}}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} + 1, 1 + \frac{c_{cs}}{c_p \cdot \tau_{\text{approx}}}\right\}$. The best length of the keep-alive window of the approximate policy would minimize the maximum of the upper bounds of the ratio of the costs of the approximate policy and the optimal policy. Mathematically, the best approximate policy keep-alive window is expressed as,

$$
\arg\min_{\tau_{\text{approx}}} \max\left\{\frac{c_p \cdot \tau_{\text{approx}}}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} + 1, 1 + \frac{c_{cs}}{c_p \cdot \tau_{\text{approx}}}\right\}
$$

We can obtain the length of the approximate keep-alive window by solving the above expression.

$$
\begin{aligned}
\frac{c_p \cdot \tau_{\text{approx}}}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} + 1 &= 1 + \frac{c_{cs}}{c_p \cdot \tau_{\text{approx}}} \\
\frac{c_p \cdot \tau_{\text{approx}}}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} &= \frac{c_{cs}}{c_p \cdot \tau_{\text{approx}}} \\
(c_p \cdot \tau_{\text{approx}})^2 &= c_{cs} \cdot (c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}) \\
\tau_{\text{approx}} &= \left(\frac{c_{cs}}{c_p} \cdot \left(\tau_{\text{opt},\mathcal{H}=\phi} + \frac{c_{cs}}{c_p}\right)\right)^{\frac{1}{2}}
\end{aligned}
$$

Substituting τ_{approx} in the expression for the upper bound of the ratio of the cost of the approximate policy to the cost of the optimal policy, we get

$$
\begin{aligned}
\frac{cost(x_m, \tau_{\text{approx}})}{cost(x_m, \tau_{\text{opt},\mathcal{H}_{m-1}})} &= 1 + \frac{c_p \cdot \tau_{\text{approx}}}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}} \\
&= 1 + \frac{\tau_{\text{approx}}}{\tau_{\text{opt},\mathcal{H}=\phi} + \frac{c_{cs}}{c_p}} \\
&= 1 + \frac{\left(\frac{c_{cs}}{c_p} \cdot \left(\tau_{\text{opt},\mathcal{H}=\phi} + \frac{c_{cs}}{c_p}\right)\right)^{\frac{1}{2}}}{\tau_{\text{opt},\mathcal{H}=\phi} + \frac{c_{cs}}{c_p}} \\
&= 1 + \left(\frac{c_{cs}}{c_p \cdot \tau_{\text{opt},\mathcal{H}=\phi} + c_{cs}}\right)^{\frac{1}{2}} \\
&= 1 + \left(\frac{1}{\frac{c_p}{c_{cs}} \cdot \tau_{\text{opt},\mathcal{H}=\phi} + 1}\right)^{\frac{1}{2}} \\
&\leq 1 + 1 = 2
\end{aligned}
$$

□

C.3 APPLICATION TO POISSON AND HAWKES PROCESSES

As previously observed, the fixed policy from Theorem 4 is independent of the process and so has a keep alive window of $\tau_{\text{fixed}} = c_{cs}/c_p$ for both Poisson and Hawkes processes. The behavior of τ_{approx} from Theorem 5 is more interesting. For Poisson processes, we know that $\tau_{\text{opt},\mathcal{H}=\phi}$ is either 0 or ∞. In the former case, $\tau_{\text{approx}} = \tau_{\text{fixed}}$ and the approximation ratio of 2 is tight. (Consider any input where $x_m > c_p/c_{cs}$.) In the latter case however, $\tau_{\text{approx}} = \infty = \tau_{\text{opt}}$ and so the approximation is 1.

For Hawkes processes, the length of the keep-alive window for the approximate policy is,

$$
\tau_{\text{approx}} = \left(\frac{c_{cs}}{c_p} \cdot \left(\tau_{\text{opt},\mathcal{H}=\phi} + \frac{c_{cs}}{c_p}\right)\right)^{\frac{1}{2}}
$$

From the more general expression for $\tau_{\text{opt},\mathcal{H}}$ for Hawkes processes,

$$
\tau_{\text{opt},\mathcal{H}=\phi} = \frac{1}{\beta} \cdot \left(\log \alpha - \log\left(\frac{c_p}{c_{cs}} - \lambda_0\right)\right)
$$

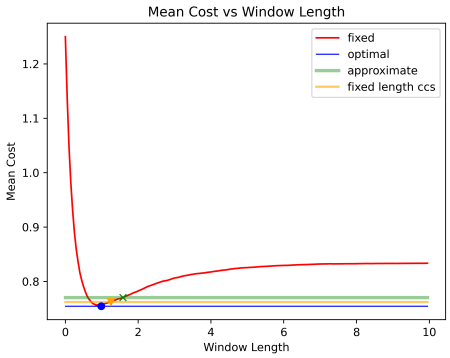
Combining these, we have

$$
\tau_{\text{approx}} = \left(\frac{c_{cs}}{c_p} \cdot \left(\frac{1}{\beta} \cdot \left(\log \alpha - \log\left(\frac{c_p}{c_{cs}} - \lambda_0\right)\right) + \frac{c_{cs}}{c_p}\right)\right)^{\frac{1}{2}}
$$

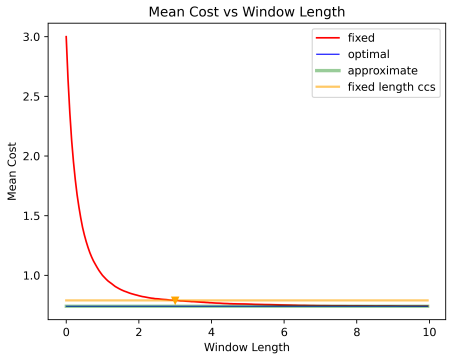
This illustrates how $\tau_{\text{opt},\mathcal{H}=\phi}$ implicitly brings the parameters of the Hawkes process into τ_{approx}.

C.4 PERFORMANCES ON SIMULATED HAWKES PROCESSES

We present similar simulations from before with two additional policies included (approximate policy and fixed policy of length c_{cs}). Generally, they demonstrate the conservative approach these policies take to achieve their worst case guarantees. In general, both perform worse than both the optimal policy (blue) and best fixed policy (red). The relative performance of

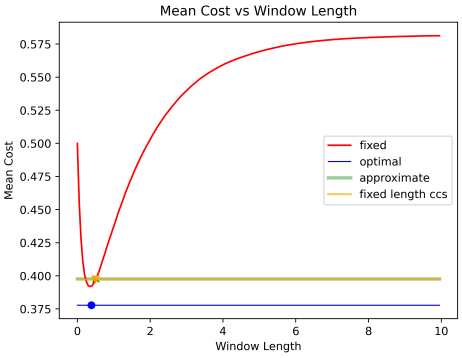


(a) $\lambda_0 = 0.6, \alpha = 1.2, \beta = 2.4, c_p = 1.0, c_{cs} = 1.25$

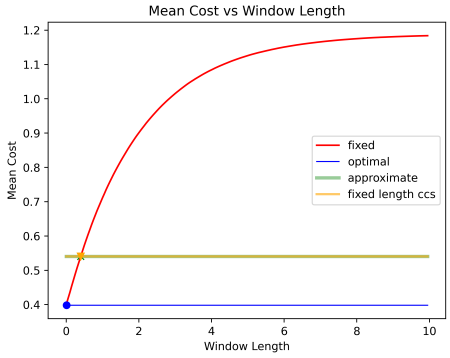


(b) $\lambda_0 = 0.45, \alpha = 0.8, \beta = 1.2, c_p = 1.0, c_{cs} = 3.0$

Figure 4: Plots of average cost comparisons between different policies for cases where $c_p \leq c_{cs}$



(a) $\lambda_0 = 0.65, \alpha = 1.4, \beta = 2.2, c_p = 1.0, c_{cs} = 0.5$

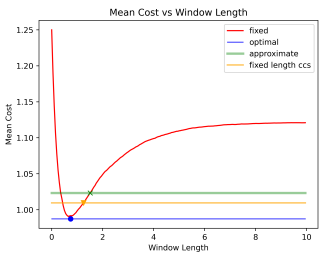


(b) $\lambda_0 = 0.5, \alpha = 0.6, \beta = 1.5, c_p = 1.0, c_{cs} = 0.4$

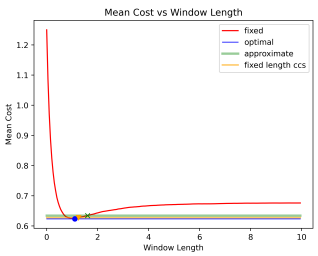
Figure 5: Plots of average cost comparisons between different policies for cases where $c_p \geq c_{cs}$

the two policies is not consistent, with each better in some cases. The gap between the yellow line and the blue line is what provided room for the improvement of Optimal-TTL policy over Fixed policy.

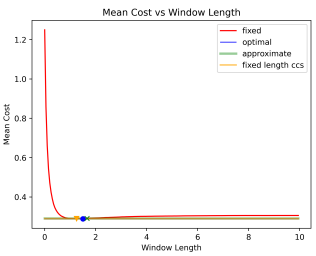
As Figures 4 and 5 illustrate, τ_{approx} is always more conservative than τ_{fixed} in that it chooses a weakly longer window length (which is is how it achieves its stronger worst case performance guarantee). Despite this, their average performance is often quite similar. In some situations, like Figure 4 (a), both policies are excessively conservative and so the extra conservatism of τ_{approx} causes it to perform worse. This effect is bounded however, because in situations like Figure 5 (b) where $\tau_{\text{opt},\mathcal{H}=\phi}$ is close to zero they become essentially the same policy. In contrast, when the optimal window length is long, like Figure 4 (b), τ_{approx} performs better. Again however the effect is small, this time because when optimal window lengths are relatively long it is typically unlikely that it will actually be a long time until the next arrival.



(a) $\alpha = 0.8$

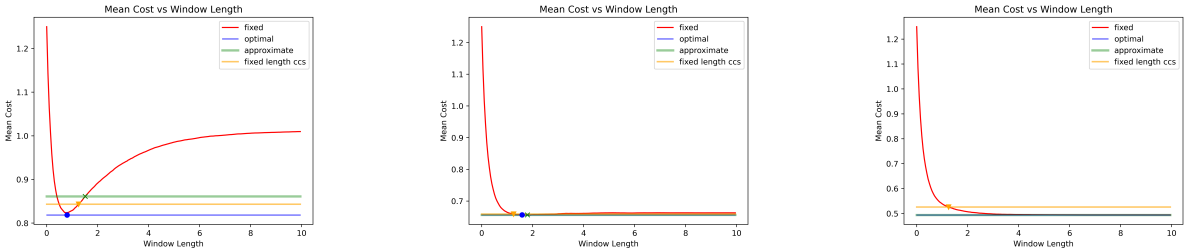


(b) $\alpha = 1.4$



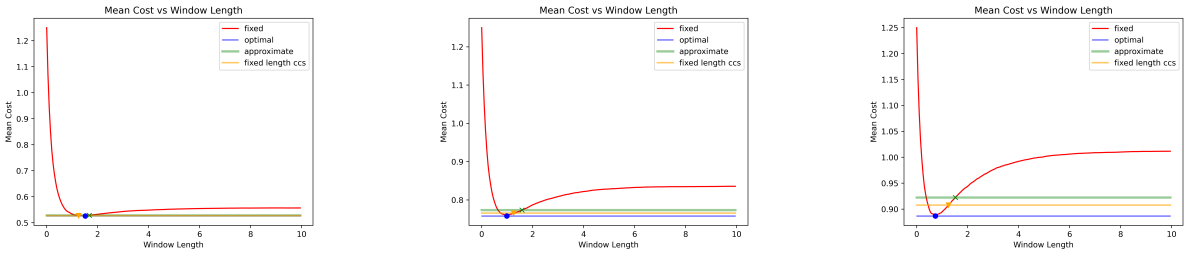
(c) $\alpha = 2.0$

Figure 6: Plots of average costs for policies when α is increased, given $\lambda_0 = 0.6, \beta = 2.4, c_p = 1.0, c_{cs} = 1.25$



(a) $\lambda_0 = 0.5$ (b) $\lambda_0 = 0.75$ (c) $\lambda_0 = 1.0$

Figure 7: Plots of average costs for policies when λ_0 is increased, given $\alpha = 1.2, \beta = 2.4, c_p = 1.0, c_{cs} = 1.25$

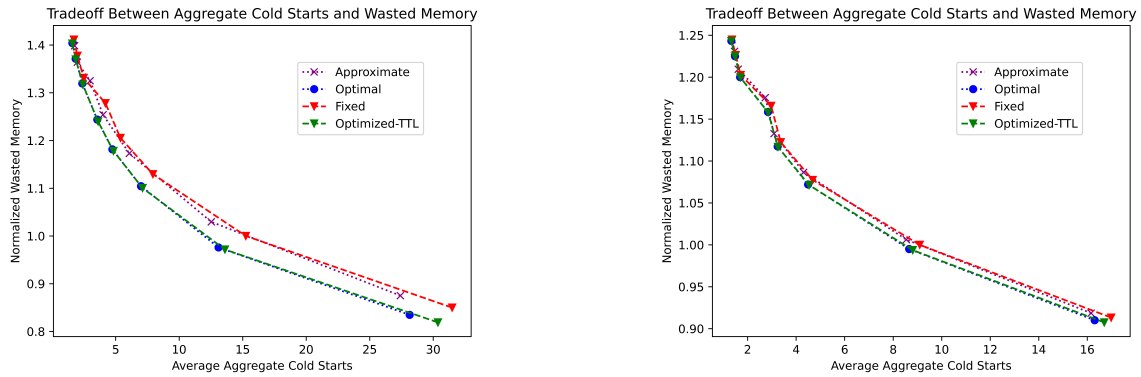


(a) $\beta = 1.8$ (b) $\beta = 2.4$ (c) $\beta = 3.0$

Figure 8: Plots of average costs for policies when β is increased, given $\lambda_0 = 0.6, \alpha = 1.2, c_p = 1.0, c_{cs} = 1.25$

C.5 AZURE DATATRACE PERFORMANCE RESULTS

Figure 9 plots the trade-off curve between the average number of cold starts per application vs the normalized wasted memory for optimal, optimized-TTL, approximate and fixed policies. In Figure 9 (a), the trade-off curve is plotted when including only those applications that follow the Hawkes process during day 9. The trade-off Pareto curve of Figure 9 (b) plots the average number of cold starts per application vs the normalized memory for all applications invoked during day 9. The plots in Figure 9 show that the trade-off Pareto curve of the approximate policy is very slightly better than the fixed policy, but substantially worse than the optimal policy, and thus Optimized-TTL as well.



(a) Evaluation only on Hawkes process applications (b) Evaluation on all applications

Figure 9: Trade-off curve of average number of cold starts vs normalized wasted memory

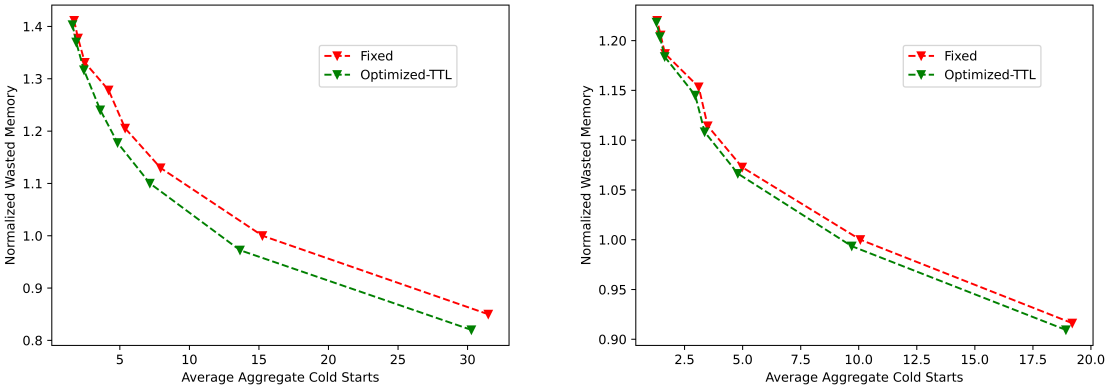
Procedure	Avg. Cold Start Savings (Hawkes)	(All)	Avg. Memory Savings (Hawkes)	(All)
Optimized-TTL (fix)	0.834	0.1393	0.043	0.0085
Optimized-TTL (no-fix)	1.037	0.0574	0.053	0.0035

Table 1: Average performance improvement over fixed policy

D USE OF SEPARATE DATA FOR GOODNESS OF FIT

The goodness of fit test is known to have a few limitations when the same data is used both to estimate the parameters and to compute the KS- statistic. Reynaud-Bouret et al. [2014] show that the Hawkes process parameters when examined for goodness of fit on the same dataset which was used for parameter estimation leads to a high bias. The authors propose sub-sampling as a reasonable solution to this problem. Rather than sub-sampling we took the advantage of additional data we are not currently using (e.g. day 7). Van Hasselt et al. [2016], Kash et al. [2019] show a similar problem and solution for training and applying double Deep Q-learning Networks (DQNs).

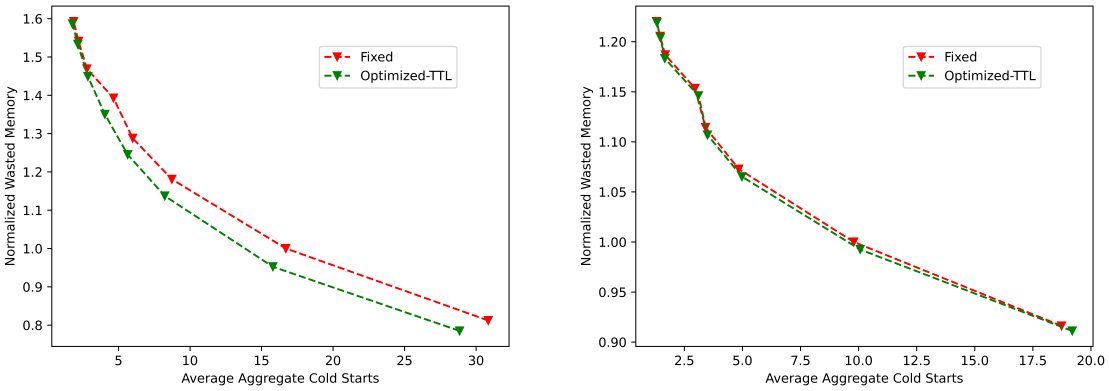
We report the results for the Optimized-TTL policy. We refer to the procedure where the goodness of fit is based on arrivals of application invocations on day 7 as "fix", whereas the procedure where the goodness of fit is based on arrivals of applications invocations on day 8 (same day as parameters estimated) is referred to as "no-fix". To compare the "fix" and "no-fix" procedures of selecting appropriate Hawkes process applications, we collect the common pool of applications invoked on day 7, day 8 and day 9. The Hawkes process applications in "fix" refer to applications where the parameters were estimated on day 8, and the KS test was performed on day 7 of the corresponding applications. The Hawkes process applications in "no-fix" refer to applications where the parameters were estimated on day 8, and the KS test was performed on the same day 8 of the corresponding applications (these are the common pool of applications present on day 7 and day 8). The number of common pool applications on day 7, day 8, and day 9 = 14788. Of these 3,694 applications fall into the 25 percentile apps that were selected as Hawkes process apps for each procedure ("fix", and "no-fix"). The amount of overlap on applications between the two tests, that is, the overlap of applications that passed the test on day 7 and applications that passed the test on day 8 = 2754. Overlap percentage = 2754/3694 = 0.745 . We show the plots of the trade-off curve between the fixed policy and the optimized-TTL policies for the overlapped apps in Figures 10, and 11 for "fix", and "no-fix" procedures. Figures 10 (a), and 11 (a) show the trade-off curves for treated apps, whereas Figures 10 (b), and 11 (b) show the trade-off curves for all apps. We compute the cold start savings as the area between the optimized-TTL curve and the fixed policy curve divided by the maximum amount of wasted memory. Similarly, the wasted memory savings is the area between the optimized-TTL curve and the fixed policy curve divided by the maximum number of average cold-starts. The corresponding versions of the cold start savings and memory savings for optimized-TTL policy are given in Table 1. The “fix” version shows a slightly weaker performance on the treated apps, but a noticeably better performance on all apps than either other version. So we view this as a demonstration that the "fix" does improve the selection of which apps to treat as Hawkes process and proceed to test the goodness of fit based on the arrivals of application invocations on day 7.



(a) Hawkes process applications

(b) All applications

Figure 10: Trade-off curve for optimized-TTL and fixed policies where goodness of fit is evaluated on day 7



(a) Hawkes process applications

(b) All applications

Figure 11: Trade-off curve for optimized-TTL and fixed policies where goodness of fit is evaluated on day 8