ONLINE CACHING WITH FETCHING COST FOR ARBITRARY DEMAND PATTERN: A DRIFT-PLUS-PENALTY APPROACH

Shashank P.

Department of Computer Science and Engineering
IIT Dharwad, Dharwad
shashankp2832@gmail.com

B. N. Bharath

Department of Electrical Engineering
IIT Dharwad, Dharwad
bharathbn@iitdh.ac.in

ABSTRACT

In this paper, we investigate the problem of caching in a single server setting from the stochastic optimization viewpoint. The goal here is to optimize the time average cache hit subject to a time average constraint on the fetching cost and the cache constraint when the demands are non-stationary and possibly correlated across time. We propose a modified Drift-Plus-Penalty (DPP) algorithm where at each time slot, we greedily minimize the sum of fetching cost and an estimated cache hit multiplied by a factor $V > 0$. Since the problem does not exhibit an equilibrium optimal solution, we use a T slot *lookahead* metric where we benchmark the performance of the proposed algorithm with respect to a genie aided cache hit which has access to demands of the future T slots. We show that with a probability of at least $1-\delta$, the cache hit of the proposed algorithm with respect to the genie scales as $\mathcal{O}\left(\frac{T^2+T\log R}{V\sqrt{R}}+\frac{T}{V}\right)+\mathtt{mse}_{R,T}$ and a fetching cost of $\mathcal{O}\left(\frac{V\log(\frac{1}{\delta})}{R}\right)$ is achievable, where $\mathtt{mse}_{R,T}$ is the Mean Squared Error (MSE) of the predictor. We make the following observations to achieve better performance: (i) the MSE of the predictor should be less, (ii) V should be chosen large to achieve better cache hit but results in higher fetching cost, (iii) higher R to compensate for larger V, i.e., more time is required to achieve lower fetching cost. We corroborate these findings using a real world dataset, and show that the proposed algorithm outperforms some well known caching algorithms.

***Index Terms*—** Caching, online learning, drift-plus-penalty algorithm, regret, prediction.

1. INTRODUCTION

The problem of caching aims to store popular contents at local caches for fast access. Increased demand for internet videos and the advent of several modern applications such as Augmented Reality (AR) and Virtual Reality (VR) that consume high bandwidth has made caching more necessary now than ever. Caching as online learning problem has been extensively studied in the past by various researchers. Performance measures such as competitive ratio and regret have been used to benchmark various caching algorithms. Using cache miss as a metric, algorithms such as First In First Out (FIFO), Least Recently Used (LRU) and Least Frequently Used (LFU) are shown to achieve optimal competitive ratio [1, 2, 3]. Competitive ratio based approach has the disadvantage that a small ratio may have large gap between the performances of the optimal policy and the proposed algorithm. A natural resolution to this is to use regret based approach.

A number of authors have devised algorithms that result in low regret. For example, the authors in [4] devised uncoded caching algorithm for a Bipartite network that achieves optimal regret. On the other hand, the authors in [5, 6] proposed Online Convex Optimization (OCO) based caching algorithms that achieve optimal regrets with (also, see [7]) and without coding. The other line of work include [8, 9, 10]. Despite its popularity, most of the regret based approaches assume adversarial setting that does not exploit the stochastic nature of the requests. There are a few exceptions to this including [11, 12]. However, these authors assume i.i.d. requests, which makes the problem less applicable and the analysis is relatively easy. Since caching involves fetching files from the server, fetching cost plays a major role. Extension to non-i.i.d. requests was undertaken in [13, 14] but did not include fetching cost. Most of the above mentioned work either fail to consider arbitrarily correlated demands or does not include fetching cost; this work aims to fill this gap.

Contributions of the paper: In this paper, we consider the problem of caching in a single server when the requests are arbitrarily correlated across time. In particular, we aim to maximize the average cache hit subject to fetching cost and cache constraints. We propose a modified version of Drift-Plus-Penalty (DPP) algorithm to solve the problem, where the predicted demands are used as a proxy in place of true demands. The DPP aims to minimize the difference of fetching cost and the instantaneous cache hit scaled by a parameter $V > 0$. This parameter can be used to balance the cache hit and the fetching cost. Due to non-stationary demands (possibly correlated), the equilibrium notion of optimal solution does not exists. To benchmark our algorithm, we use the approach taken in [15]. We divide time into R blocks of T slots each. We benchmark our algorithm with a genie aided policy which assumes that at the beginning of block r, it has access to the future demands in slots $r+1, r+2, \ldots, r+T$, i.e., the next block.

We show that with a probability of at least $1-\delta$, the proposed algorithm (DPP-Cache) achieves a regret (with respect to the genie) which scales as $\mathcal{O}\left(\frac{T^2+T\log R}{V\sqrt{R}}+\frac{T}{V}\right)+\texttt{mse}_{R,T}$ and a fetching cost of $\mathcal{O}\left(\frac{V\log(\frac{1}{\delta})}{R}\right)$, where $\texttt{mse}_{R,T}$ is the Mean Squared Error (MSE) of the predictor. Higher T results in a higher cache hit for the genie aided scenario but also increases the regret. To handle this, either R or V should be increased. Increasing V results in increased fetching cost leading to $1/V$-V tradeoff. On the other hand increasing R results in better cache hit as well as lower fetching cost but requires more time. We use real world data set () to show that the DPP-Cache either outperforms existing algorithms such as FTPL, LFU while requiring less fetching cost or requires very less fetching cost (download rate) to achieve considerable fraction of the cache hit achieved by LRU and LeadCache.

2. CACHING MODEL AND PROBLEM SETUP

In this paper, we consider the problem of caching in a single server with users requesting files from a catalog $\mathcal{F} := \{1, 2, \ldots, F\}$. We assume that the time is slotted, and the demands (the total number of requests) is arbitrarily correlated across time slots. Let us denote the demand for file $f \in \mathcal{F}$ in the time slot t by $\theta_{f,t} \in \mathbb{R}^+$, and the corresponding demand vector by $\Theta_t := (\theta_{1,t}, \theta_{2,t}, \ldots, \theta_{1,F})^T$. The server is capable of storing at most C files in its local cache. Let us use $z_{f,t} = 1$ to indicate that file f is stored in the local cache at time t, and zero otherwise. The corresponding vector be denoted by $Z_t := (z_{1,t}, z_{2,t}, \ldots, z_{F,t})^T$. The cache capacity imposes the following constraint $\mathbf{1}^T Z_t = C$. Further, we assume that the cache is refreshed at the beginning of every time slot. Since storing new files requires the server to fetch it from a server; this amounts to a fetching cost, which needs to be controlled. Note that the fetching cost in terms of the number of new files being cached is given by $\frac{||z_t - z_{t-1}||_1}{2}$ (see [16]). Now, each server wishes to maximize the "average" cache hit subject to cache and fetching cost constraints. Mathematically, we have

$$\begin{aligned}
\texttt{Problem 0:} \quad & \max_{Z_t : t \in \mathbb{N}} \lim_{\tau\to\infty} \frac{1}{\tau}\sum_{t=0}^{\tau-1} \Theta_t^T Z_t \\
\text{subject to} \quad & \lim_{\tau\to\infty} \frac{1}{\tau}\sum_{t=0}^{\tau-1} ||Z_t - Z_{t-1}||_1 \le \nu \\
& \mathbf{1}^T Z_t = C \text{ and } Z_t \in \{0,1\}^F,
\end{aligned}$$

where the last constraint should be satisfied for all t, and $\nu > 0$. Benchmarking our algorithm with respect to the solution of $\texttt{Problem 0}$ has two issues. First, the problem is combinatorial, and hence NP-hard [17]. Second, the demands can vary arbitrarily, and hence the equilibrium notion of optimal solution for the $\texttt{Problem 0}$ does not exists [18]. The first issue can be resolved by replacing the discrete caching variable $Z_t \in \{0,1\}^F$ by $X_t \in [0,1]^F$. Let the corresponding relaxed problem be denoted by $\texttt{Problem (1)}$. As in [18], the second issue will be handled by resorting to T-slot look a head solution, which is explained next. Consider two integers R and T, and divide the RT time slots into R blocks of size T slots each. In particular, we benchmark our proposed algorithm with the solution for the following problem obtained by a genie who at the beginning of each block $r \in \{0, 1, \ldots, R-1\}$ has access to the future demands, i.e., Θ_t for $t = rT+1, rT+2, \ldots (r+1)T$

$$\begin{aligned}
\texttt{Problem (*):} \quad & \max_{X_t} \frac{1}{T}\sum_{t=rT+1}^{(r+1)T} \Theta_t^T X_t \\
\text{subject to} \quad & \frac{1}{T}\sum_{t=rT+1}^{(r+1)T} ||X_t - X_{t-1}||_1 \le \nu \\
& \mathbf{1}^T X_t = C \text{ and } X_t \in [0,1]^F.
\end{aligned}$$

Clearly, there is a trade-off between T and R. Higher T makes the problem closer to $\texttt{Problem 0}$ but may be significantly better than the performance of the proposed algorithm. Thus, T and R can be used as a tuning parameters to understand the performance of the proposed algorithm better. Let X_t^* be the optimal solution to $\texttt{Problem (*)}$. Thus, within the block r, the optimal cache hit is given by $\Phi_r^* := \frac{1}{T}\sum_{t=rT+1}^{(r+1)T} \Theta_t^T X_t^*$, and the overall cache hit is $\Phi^* = \lim_{R\to\infty} \frac{1}{R}\sum_{r=0}^{R-1} \Phi_r^*$. Now, we state a few minor assumptions.

Assumption 1 *We assume that the demands are bounded in every slot t, i.e.,* $\theta_{i,t} \le d_{max}$. *Further* $\delta_t \le \delta_{max}$.

In the following subsection, we present the algorithm based on drift-plus-penalty, and subsequently prove its guarantees.

2.1. DPP Caching (DPP-cache) Algorithm

In this paper, we take the DPP based approach for solving stochastic optimization problem. In this approach, at each time slot, a combination of the cache hit and the drift is minimized. Towards stating the DPP-cache algorithm, we first consider the following virtual queue that measures the extent of fetching cost violation

$$Q(t+1) := \max\{Q(t) + \delta_t ||X_t - X_{t-1}||_1 - \nu, 0\}. \quad (1)$$

The Lyapunov drift is defined as [18]

$$\mathcal{L}(t) := \frac{Q^2(t) - Q^2(t-1)}{2}. \quad (2)$$

Algorithm 1: DPP-Cache

```
RANDOMLY INITIALISE Θ_0, X_0 ∈ R^F and Q(t) = 0
for t = 0, 1, ..., RT do
    USE THE PREVIOUS DEMANDS Θ_{-t}, AND PREDICT
      Θ̂_t = f_{w_t}(Θ_{-t}) // This depends on the predictor
      (eg. neural network) used.
    SOLVE THE FOLLOWING DPP PROBLEM TO GET X_t:
      arg min_{X: 1^T X = C, X ≽ 0} [Q(t)Δ(t) − V Θ̂_t^T X]
    UPDATE THE CACHE USING X_t
    UPDATE THE QUEUE AT THE END OF TIME SLOT t:
      Q(t) = max{Q(t−1) + δ_t ||X_t − X_{t−1}||_1 − ν, 0}
end
```

The goal of a typical DPP algorithm is to greedily minimize $\mathcal{L}(t) - V\Theta_t^T X_t$ with respect to X_t subject to the cache constraint, where $V > 0$ is the parameter that balances the cache hit and the fetching cost violation. Unfortunately, the demand in the slot t is unknown, and hence need to be predicted using either classical approaches or using neural network based models. We assume that $\hat{\Theta}_t$ is an estimate of Θ_t at the beginning of slot t based on past observation. The details about the prediction methodology, and its impact on the performance will be presented in the later sections. Using this estimate, the DPP-cache minimizes an upper bound on the DPP

$$\begin{aligned}\mathcal{L}(t) - V\hat{\Theta}_t^T X_t &\le \frac{\Delta^2(t)}{2} + Q(t)\Delta(t) - V\hat{\Theta}_t^T X_t \\ &\le \frac{B^2}{2} + Q(t)\Delta(t) - V\hat{\Theta}_t^T X_t, \quad (3)\end{aligned}$$

where $B := \sqrt{4C^2 + \nu^2}$, and $\Delta(t) := ||Z_t - Z_{t-1}||_1 - \nu$. In the above, the first inequality follows from using the fact that $Q^2(t) \le (Q(t) + \Delta(t))^2$, and the second inequality follows from the fact that $||Z_t - Z_{t-1}||_1 \le 2C$. The DPP-cache solves the right hand side of the above equation at each time slot, as shown in Algorithm 1. In the algorithm, Θ_{-t} refers to all demand vectors prior to the slot t. Depending on the prediction procedure, this may vary. Finally, X_t is rounded to get an integer solution. An extension of this to more sophisticated techniques such as pipage rounding will be considered as future work, and is ignored in our analysis. In the next section, we provide a theoretical guarantee of the proposed online caching algorithm (DPP-cache).

3. PERFORMANCE GUARANTEES OF DPP-CACHE

First, we show that $\sum_{t=0}^{RT-1} \Theta_t^T X_t \ge \sum_{t=0}^{RT-1} \mathbb{E}\left[\Theta_t^T X_t | \mathcal{F}_r\right] -$ `error` is achieved by the DPP-Cache with high probability. Here, $\mathcal{F}_r$ is the information until the previous block r, and the `error` is shown to depend on T, R, V and other parameters of the caching system. We use the following definition of the MSE for stating the main result:

$$\texttt{mse}_{R,T} := \frac{1}{RT}\sum_{r=0}^{R}\sum_{t=rT+1}^{(r+1)T}\left(\sqrt{\mathbb{E}\left[E_t|\mathcal{F}_{r-1}\right]} + \sqrt{E_t}\right) \quad (4)$$

where $E_t := ||\hat{\Theta}_t - \Theta_t||^2$. We state our first main result below whose proof is provided in Sec. 8.

Theorem 1 *For* $V \ge \max\left\{\frac{\nu(\delta_{max}C+\nu)}{4Cd_{max}}, \frac{\nu^2}{2d_{max}C}\right\}$ *and* $c_1 = \frac{1}{\alpha}\log\frac{2DR}{\delta}$*, the DPP-cache achieves the following with a probability of at least* $1 - \frac{\delta}{2}$*:*
(1) The average loss

$$\Phi_{loss} \le \Delta_{T,R,V} + \frac{\alpha B^2 + 2C\delta_{max}\log\frac{2DR}{\delta}}{2\alpha V} + \sqrt{C}\texttt{mse}_{R,T},$$

where

$$\Phi_{loss} := \frac{1}{RT}\sum_{r=0}^{R}\sum_{t=rT+1}^{(r+1)T}\left[\mathbb{E}\left[\Theta_\tau^T X_\tau^* | \mathcal{F}_{r-1}\right] - \Theta_t^T X_t\right],$$

$\texttt{mse}_{R,T}$ *is as defined in* (4)*, and the error term*

$$\Delta_{T,R,V} := \sqrt{\frac{2\beta_T^2\log\frac{2}{\delta}}{RT^2V^2}} + \frac{\delta_{max}^2 C^2 T + q_{max}^2 T}{2V}. \quad (5)$$

Here, $q_{max} := (\delta_{max}C + \nu)$*,* $\beta_T := \delta_{max}CTc_1 + \Psi_{T,C}$*, and*

$$\Psi_{T,C} := \frac{4Td_{max}VC + \delta_{max}^2 C^2 T^2 + (\delta_{max}C + \nu)^2 T^2}{2}.$$

(2) The average fetching cost satisfies

$$F_{R,T} \le \frac{1}{RT}\log\frac{2D}{\delta}, \quad (6)$$

where $F_{R,T} := \frac{1}{RT}\sum_{t=1}^{RT-1}\left[\delta_t||X_{rT-1} - X_{t-1}|| - \nu\right]$ *and* $D := \left(\frac{e^{\alpha T(\delta_{max}C+\nu)}}{1-\rho} - \frac{\rho}{1-\rho}\right)e^{\frac{4\alpha V d_{max}CT}{\nu}}$*. Here*

$$\alpha := \frac{3(T-1)\nu}{3T^2(\delta_{max}+\nu) + T\nu(\delta_{max}+\nu)(T-1)},$$

and

$$\rho = 1 - \frac{3(T-1)^2\nu^2}{6T^2(\delta_{max}+\nu)^2 + 2T\nu(\delta_{max}+\nu)(T-1)}.$$

Note that $\alpha \sim \mathcal{O}(\frac{1}{T})$ and therefore, for sufficiently large T, $\rho \in [0, 1]$, and $1 - \rho \sim \mathcal{O}(1)$. Now, it is easy to see that $D \sim \mathcal{O}(\exp\{\alpha VT^2\})$. Using these, the fetching cost constraint is given by $\frac{1}{RT}\log\frac{2D}{\delta} \sim \mathcal{O}(\frac{V\log(\frac{1}{\delta})}{R})$. Further, $c_1 \sim \mathcal{O}(T(VT + \log R))$ and $\Psi_{T,C} \sim \mathcal{O}(TV + T^2)$ imply

$\beta_T \sim \mathcal{O}(T^2(VT + \log R))$. Using these, we have $\Delta_{T,R,V} \sim \mathcal{O}\left(\frac{T^2+T\log R}{V\sqrt{R}} + \frac{T}{V}\right)$. This implies the following corollary.

Corollary 1 *With a probability of at least* $1-\delta$, *we have*

$$\Phi_{loss} \leq \mathcal{O}\left(\frac{T^2 + T\log R}{V\sqrt{R}} + \frac{T}{V}\right) + \mathtt{mse}_{R,T} \qquad (7)$$

and the fetching cost satisfies

$$\frac{1}{RT}\sum_{t=1}^{RT-1}[\delta_t||X_{rT-1} - X_{t-1}|| - \nu] \leq \mathcal{O}\left(\frac{V\log(\frac{1}{\delta})}{R}\right).$$

Remark: As expected the DPP-Cache results in a better cache hit performance provided the demand prediction is accurate, i.e., $\mathtt{mse}_{R,T}$ is small as shown in the above result. In particular, the algorithm used to predict the demands plays a major role in the overall performance of the algorithm. For a fixed R and T, higher values of V results in a better cache hit performance but results in a poor fetching cost; this depicts the $1/V$-V trade-off [19]. Alternatively, if T is large, then the genie aided cache hit is large. If the upper bound on Φ_{loss} is small for larger T, then the DPP-Cache results in a better cache hit. However, larger T increases the bound on Φ_{loss}, and hence to handle this, either V or R should be increased. Higher V results in the trade-off mentioned above while large values of R results in longer time for the DPP-Cache to achieve better performance. Some of these observations are corroborated in the following experimental results section.

4. EXPERIMENTAL RESULTS

In this section, we corroborate our theoretical findings by evaluating the performance of the DPP-Cache algorithm on real world datasets.

Dataset and Neural Network Description: The dataset used is "311 Service Requests Pitt" from Kaggle.[1]. It consists of public service requests such as "potholes, abandoned vehicles, building maintenance etc.," their description, and the time-stamps of the requests from Feb. 2017 to Nov. 2018, which is divided into 200 time slots. To use this data for caching scenario, we replace the type of service as a proxy for the files. There are a total of 423 unique files (services) and around 6 million requests. This leads to an average of 3300 requests per slot. At the beginning of every slot, a neural network is updated based on the demands from the past 5 time slots, and is subsequently used to predict the next set of demands using the past 3 slots demands as input. The neural network consists of six LSTM layers and two dense layers (number of output is equal to 423). This amounts to a total of 27 million neural coefficients.

Performance Results: The cache is updated using Algorithm 1 at the beginning of every slot. We have used $C = 42$, and the parameter V is increased across time slot t as $\mathcal{O}(\sqrt{t})$. The cost constraint was set to $\nu = 20$. As in [16], we have used cache hit as the metric, and the download rate for measuring the fetching cost. For simplicity, we have assumed $\delta_t = 1$ for all t. Figs. 1-2 show the plots of cache hit and download rate as a function of time slot for different caching schemes such as LRU, LFU, FTPL and DPP-Cache. It is clear from the figures that DPP-Cache eventually (after 120 slots) achieves better cache hit than LFU and FTPL, while achieving a cache hit comparable to LRU. However, the fetching cost of DPP-Cache is significantly less compared to LRU thereby demonstrating superiority of the proposed algorithm.

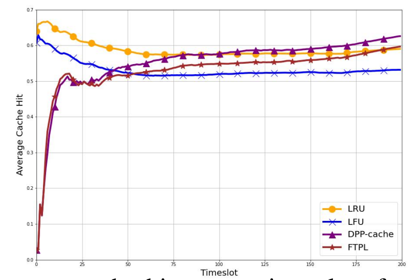

Fig. 1: Average cache hit versus time slots for LRU, LFU, DPP-Cache and FTPL.

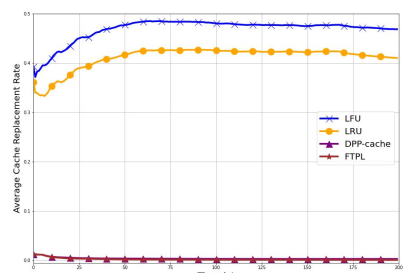

Fig. 2: Average download rate versus time slot for LRU, LFU, DPP-Cache and FTPL.

5. CONCLUSIONS

In this paper, we considered the problem of caching in a single server setting from the stochastic optimization viewpoint. With the goal of maximizing the time average cache hit subject to a time average constraint on the fetching cost, we proposed a modified Drift-Plus-Penalty (DPP) algorithm where at each time slot, we greedily minimize the sum of fetching cost and an estimate cache hit multiplied by a factor $V > 0$. Using T slot *lookahead* metric as a benchmark on the performance of the proposed algorithm, we proved a high probability bound on the difference in the cache hit of the proposed algorithm with respect to the genie. In particular, the bound demonstrated the trade-off between performance and the fetching cost. Finally, we used the real world data set to show that the DPP-Cache outperforms some of the well known caching algorithms.

[1]Link: dataset

6. REFERENCES

[1] D. Lee, J. Choi, J.-H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim, "On the existence of a spectrum of policies that subsumes the least recently used (lru) and least frequently used (lfu) policies," *SIGMETRICS Perform. Eval. Rev.*, vol. 27, no. 1, p. 134–143, may 1999. [Online]. Available: https://doi.org/10.1145/301464.301487

[2] A. Dan and D. Towsley, "An approximate analysis of the lru and fifo buffer replacement schemes," *SIGMETRICS Perform. Eval. Rev.*, vol. 18, no. 1, p. 143–152, apr 1990. [Online]. Available: https://doi.org/10.1145/98460.98525

[3] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, p. 836–845, apr 2016.

[4] D. Paria and A. Sinha, "$\texttt{LeadCache}$: Regret-optimal caching in networks," in *Advances in Neural Information Processing Systems*, 2021. [Online]. Available: https://openreview.net/forum?id=yrmvFIh_e5o

[5] R. Bhattacharjee, S. Banerjee, and A. Sinha, "Fundamental limits on the regret of online network-caching," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 4, no. 2, jun 2020. [Online]. Available: https://doi.org/10.1145/3392143

[6] G. S. Paschos, A. Destounis, L. Vigneri, and G. Iosifidis, "Learning to cache with no regrets," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. IEEE Press, 2019, p. 235–243. [Online]. Available: https://doi.org/10.1109/INFOCOM.2019.8737446

[7] N. Garg, M. Sellathurai, V. Bhatia, B. N. Bharath, and T. Ratnarajah, "Online content popularity prediction and learning in wireless edge caching," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1087–1100, 2020.

[8] D. Paria and A. Sinha, "$\texttt{LeadCache}$: Regret-optimal caching in networks," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 4435–4447. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/2387337ba1e0b0249ba90f55b2ba2521-Abstract.html

[9] X. Xu, M. Tao, and C. Shen, "Collaborative multi-agent multi-armed bandit learning for small-cell caching," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2570–2585, 2020.

[10] T. Si Salem, G. Neglia, and S. Ioannidis, "No-Regret Caching via Online Mirror Descent," in *ICC 2021 - IEEE International Conference on Communications*, Montreal (virtual), Canada, Jul. 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03376156

[11] P. Blasco and D. Gündüz, "Multi-armed bandit optimization of cache content in wireless infostation networks," in *2014 IEEE International Symposium on Information Theory*, 2014, pp. 51–55.

[12] A. Bura, D. Rengarajan, D. Kalathil, S. Shakkottai, and J.-F. Chamberland, "Learning to cache and caching to learn: Regret analysis of caching algorithms," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, p. 18–31, feb 2022. [Online]. Available: https://doi.org/10.1109/TNET.2021.3105880

[13] S. O. Somuyiwa, A. György, and D. Gündüz, "Multicast-aware proactive caching in wireless networks with deep reinforcement learning," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.

[14] B. N. Bharath, K. G. Nagananda, D. Gündüz, and H. V. Poor, "Caching with time-varying popularity profiles: A learning-theoretic perspective," *IEEE Transactions on Communications*, vol. 66, no. 9, pp. 3837–3847, 2018.

[15] M. Neely, "Universal scheduling for networks with arbitrary traffic, channels, and mobility," *Proceedings of the IEEE Conference on Decision and Control*, 01 2010.

[16] S. Mukhopadhyay and A. Sinha, "Online caching with optimal switching regret," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 1546–1551.

[17] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[18] X. Wei, H. Yu, and M. J. Neely, "A probabilistic sample path convergence time analysis of drift-plus-penalty algorithm for stochastic optimization," *arXiv preprint arXiv:1510.02973*, 2015.

[19] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

7. USEFUL LEMMA

Lemma 1 *The drift in the queue is bounded, i.e.,* $|Q(t) - Q(t-\tau)| \leq \tau(2\delta_{max}C + \nu)$.

Proof: By the definition of the queue update, we have

$$\begin{aligned} Q(t) &= \max\{Q(t-1) + ||X_t - X_{t-1}||_1 - \nu, 0\} \\ &\leq Q(t-1) + |\delta_t||X_t - X_{t-1}||_1 - \nu| \\ &\leq Q(t-\tau) + \sum_{l=1}^{\tau-1} |\delta_{t-l}||X_{t-l} - X_{t-l-1}||_1 - \nu|, \end{aligned}$$

where the last inequality follows from induction. Using triangle inequality, and the fact that $\delta_t \leq \delta_{\max}$ for all t, and $||X_{t-l} - X_{t-l-1}||_1 \leq 2C$, we have $|\delta_{t-l}||X_{t-l} - X_{t-l-1}||_1 - \nu| \leq 2\delta_{\max}C + \nu$

8. PROOF OF THE MAIN RESULT

First, we provide the outline of the proof, which follows along the lines of [18]. Consider the following sequence:

$$X_T[r] := \sum_{t=rT+1}^{(r+1)T} \left[Q(t)\Delta(t) - V\hat{\Theta}_t^T X_t\right] + \mathcal{D}_r, \quad (8)$$

where $\Delta(t) := \delta_t ||X_t - X_{t-1}||_1 - \nu$, and

$$\begin{aligned} \mathcal{D}_r &:= V \sum_{\tau=rT+1}^{(r+1)T} \mathbb{E}\left[\hat{\Theta}_\tau^T X_\tau^* | \mathcal{F}_{r-1}\right] - C\delta_{\max} T Q(rT) \\ &- \frac{\delta_{\max}^2 C^2 T^2 + (\delta_{\max}C + \nu)^2 T^2}{2}, \end{aligned} \quad (9)$$

where $\mathcal{F}_{r-1} = \sigma(\Theta_{-(rT)})$, the sigma algebra generated by all the demands until the time slot rT. We first show the following result whose proof can be found in Sec. 9.

Lemma 2 *The sequence* $Y[R] := \sum_{r=0}^{R} X_T[r]$ *is a super Martingale.*

Since there is no guarantee that $Q(t)$ is bounded, one cannot apply large deviation bound on $\Pr\{\sum_{r=0}^{R-1} Y[R] > \lambda\}$ as the super Martingale is not bounded. In order to resolve this, we use the stopped sequence $Z[R] := Y[R \wedge \tau]$, where the stopping time

$$\tau := \inf\{t : Q(t) \geq c_1\} \quad (10)$$

for any $c_1 > 0$, which will be chosen later according to our convenience. In order to proceed further, we first state and prove the result below.

Lemma 3 *The sequence* $Z[R] = Y[R \wedge \tau]$ *defined above satisfies the following.*

- $Z[R]$ *is a super Martingale.*
- *The difference of the sequence is bounded for all* R, *i.e.,* $|Z[R+1] - Z[R]| \leq \beta_T$, *where* β_T *is as defined in Theorem 1.*

Proof: See Sec. 10.

Next, we upper bound $\Pr\{\sum_{r=0}^{R-1} X_T[r] > \lambda\}$ in terms of $\Pr\{\sum_{r=0}^{R-1} Z[R] > \lambda\}$ plus an extra term. The term $\Pr\{\sum_{r=0}^{R-1} Z[R] > \lambda\}$ can be bounded easily as $Z[R]$ is a bounded super Martingale. Defining $\mathcal{B}_{r,T} := \{Q(rT) > c_1\}$ for some $c_1 > 0$ (to be chosen later), and using Lemma 4 of [18], we get

$$\begin{aligned} \Pr\{Y[R] > \lambda\} &\leq \Pr\{Z[R] > \lambda\} + \sum_{r=1}^{R} \Pr\{\mathcal{B}_{r,T}\} \\ &\leq \Pr\{Z[R] > \lambda\} + \sum_{r=1}^{R} \mathbb{E}\{e^{\alpha Q(rT)}\} e^{-\alpha c_1} \end{aligned} \quad (11)$$

where the last inequality follows from the Markov inequality. Note that if $\mathbb{E}\{e^{\alpha Q(rT)}\} \leq D$ a quantity independent of r, then, the above can be further upper bounded. To complete the proof, we show that the queue has a bounded difference with negative drift when $Q(t)$ is large.

Theorem 2 *For* $V \geq \frac{\nu^2}{2d_{max}C}$, *the conditional average difference of the queue satisfies the following*

$$\mathbb{E}\left[Q((r+1)T) - Q(rT) | Q(rT)\right] = \begin{cases} b & \text{if } Q(rT) \leq \mathcal{T} \\ \zeta & \text{if } Q(rT) > \mathcal{T}, \end{cases}$$

where $b := T(\delta_{max}C + \nu)$, $\zeta := -(T-1)\nu$, *and* $\mathcal{T} := \frac{4Vd_{max}CT}{\nu}$.

Proof: See Sec. 11.

Using Lemma 6 of [18], we have the following Proposition.

Proposition 1 *The moments of the Queue is bounded, i.e.,* $\mathbb{E}Q(rT) \leq D$ *for all* r *and* T *provided* $V \geq \frac{\nu(\delta_{max}C+\nu)}{4Cd_{max}}$, *where* D, α *and* ρ *are as defined in Theorem 1.*

Proof: See Sec. 12.

Remark: Recall that $\alpha = \mathcal{O}(\frac{1}{T})$, and $\rho \in [0, 1]$. The bound D scales exponentially with T. Later we show that this exponential dependency can be controlled by using large enough values for R.

Using Lemma 13 of [18] for the first term in (11), and using the moment bound for the second term, we get

$$\Pr\{Y[R] > \lambda\} \leq \exp\left\{-\frac{\lambda^2}{2Tc_2^2}\right\} + RDe^{-\alpha c_1}. \quad (12)$$

Thus, by choosing $\lambda = \sqrt{2R}\beta_T\sqrt{\log\frac{2}{\delta}}$, we can guarantee that $\exp\left\{-\frac{\lambda^2}{2Tc_2^2}\right\} \leq \delta/2$. Further, $c_1 = \frac{1}{\alpha}\log\frac{2DR}{\delta}$ ensures that $RDe^{-\alpha c_1} \leq \delta/2$. This implies that with a probability of at least $1-\delta$, $Y[R] \leq \sqrt{2R}\beta_T\sqrt{\log\frac{2}{\delta}}$. This results in the following theorem.

Theorem 3 *For* $V \geq \max\left\{\frac{\nu(\delta_{\max}C+\nu)}{4Cd_{\max}}, \frac{\nu^2}{2d_{\max}C}\right\}$ *and* $c_1 = \frac{1}{\alpha}\log\frac{2DR}{\delta}$, *with a probability of at least* $1-\delta$, *the caching strategy generated by the DPP-cache satisfies the following*

$$\hat{\Phi}_{loss} \leq \Delta_{T,R,V} - \frac{1}{RTV}\sum_{r=0}^{R}\sum_{\tau=rT+1}^{(r+1)T} Q(\tau)\Delta(\tau),$$

where

$$\hat{\Phi}_{loss} := \frac{1}{RT}\sum_{r=0}^{R}\sum_{t=rT+1}^{(r+1)T}\left[\mathbb{E}\left[\hat{\Theta}_\tau^T X_\tau^* | \mathcal{F}_{r-1}\right] - \hat{\Theta}_t^T X_t\right],$$

and the error term $\Delta_{T,R,V}$ *as defined in* (5) *of Theorem 1.*

Next we present high probability guarantee on the cache hit and the fetching cost constraint satisfaction.

8.1. Average cache hit

Now, summing (2) over all RT slots, we get

$$Q^2(RT) \leq \frac{B^2}{2} + \frac{1}{RT}\sum_{r=0}^{R}\sum_{t=rT+1}^{(r+1)T} Q(t)\Delta(t). \tag{13}$$

Adding $\hat{\Phi}_{loss}$ on both sides, we get $Q^2(RT) + \hat{\Phi}_{loss} \leq \frac{B^2}{2} + \hat{\Phi}_{loss} + \frac{1}{RT}\sum_{r=0}^{R}\sum_{t=rT+1}^{(r+1)T} Q(t)\Delta(t)$. Rearranging, and using Theorem 3, we get the following bound which occurs with a probability of at least $1-\delta$

$$\hat{\Phi}_{loss} \leq \Delta_{T,R,V} + \frac{B^2}{2V} + \frac{C\delta_{\max}}{V} Q_{avg}^{R,T}, \tag{14}$$

where $Q_{avg}^{R,T} := \frac{1}{R}\sum_{r=0}^{R} Q(rT)$. Let us denote the above event by Γ_{g}. We have

$$\begin{aligned} 1-\delta &\leq \Pr\{\Gamma_{\mathrm{g}} \bigcap Q_{avg}^{R,T} < \Psi\} + \Pr\{\Gamma_{\mathrm{g}} \bigcap Q_{avg}^{R,T} > \Psi\} \\ &\leq \Pr\{\Gamma_{\mathrm{g}} \bigcap Q_{avg}^{R,T} < \Psi\} + \Pr\{Q_{avg}^{R,T} > \Psi\}. \end{aligned} \tag{15}$$

The second term above can be bounded as

$$\begin{aligned} \Pr\{Q_{avg}^{R,T} > \Psi\} &\leq \sum_{r=0}^{R}\Pr\{Q(rT) > \Psi\} \\ &\leq \sum_{r=0}^{R}\mathbb{E}e^{\alpha Q(rT)}e^{-\alpha\Psi} \leq DRe^{-\alpha\Psi} \end{aligned} \tag{16}$$

Choosing $\Psi = \frac{1}{\alpha}\log\frac{2DR}{\delta}$, we get $\Pr\{Q_{avg}^{R,T} > \Psi\} \leq \frac{\delta}{2}$. The second term in (15) can be bounded as follows

$$\begin{aligned} \Pr\{\Gamma_{\mathrm{g}} \bigcap Q_{avg}^{R,T} < \Psi\} &\leq \Pr\left\{\hat{\Phi}_{loss} \leq \Delta_{T,R,V} + \frac{B^2}{2V}\right. \\ &+ \left.\frac{C\delta_{\max}\log\frac{2DR}{\delta}}{\alpha V}\right\}. \end{aligned} \tag{17}$$

Using the above two bounds in (15), we have the following bound with a probability of at least $1-\frac{\delta}{2}$

$$\hat{\Phi}_{loss} \leq \Delta_{T,R,V} + \frac{B^2}{2V} + \frac{C\delta_{\max}\log\frac{2DR}{\delta}}{\alpha V}. \tag{18}$$

Now, it remains to bound $\mathcal{E}_{\text{error}}$ in terms of Θ_t. Towards this, we have $\mathbb{E}\left[\hat{\Theta}_t^T X_t^* | \mathcal{F}_{r-1}\right] \geq \mathbb{E}\left[\Theta_t^T X_t^* | \mathcal{F}_{r-1}\right] - \sqrt{C\mathbb{E}\left[||\hat{\Theta}_t - \Theta_t||^2 | \mathcal{F}_{r-1}\right]}$, where the above inequality follows from adding and subtracting Θ_t, using Cauchy-Shwartz inequality, and the fact that $||X_t^*|| \leq C$. Similarly, the second term can be lower bounded as $-\hat{\Theta}_t^T X_t \geq -\Theta_t^T X_t - ||\Theta_t - \hat{\Theta}_t||\sqrt{C}$. Further, using Jensen's inequality, we can write $\frac{1}{RT}\sum_{r=0}^{R}\sum_{t=rT+1}^{(r+1)T}||\Theta_t - \hat{\Theta}_t|| \leq \sqrt{\frac{1}{RT}\sum_{r=0}^{R}\sum_{t=rT+1}^{(r+1)T}||\Theta_t - \hat{\Theta}_t||^2}$. Using these in Theorem 3, we get the first part of the result in Theorem 1.

8.2. Average fetching cost constraint

From the queue update equation (1), we get

$$\begin{aligned} Q(RT) &\geq Q(RT-1) + \delta_{RT-1}||X_{RT-1} - X_{RT-2}|| - \nu \\ &\geq \sum_{t=1}^{RT-1}\left[\delta_{RT-1}||X_{RT-1} - X_{RT-2}|| - \nu\right], \end{aligned} \tag{19}$$

where the above follows from induction, and the fact that $Q(0) = 0$. Thus, $\frac{1}{RT}\sum_{t=1}^{RT-1}\left[\delta_t||X_{rT-1} - X_{t-1}|| - \nu\right] \leq \frac{Q(RT)}{RT}$. Note that using moment bound on the queue, we have $\Pr\{Q(RT) > c_1\} \leq \frac{\delta}{2}$ provided $c_1 = \log\frac{2D}{\delta}$. Thus, the average fetching cost violation is small with a probability of at least $1-\delta/2$, as given in (6). This completes the proof.

9. PROOF OF LEMMA 2

Since $\mathbb{E}\left[Y[R]|\mathcal{F}_{R-1}\right] = Y[R-1] + \mathbb{E}[X_T[R]|\mathcal{F}_{R-1}]$, to show that $Y[R]$ is a super Martingale, it suffices to show that $\mathbb{E}\left[X_T[r]|\mathcal{F}_{r-1}\right] \leq 0$. Towards this, consider

$$\begin{aligned} X_T[r] \leq& \sum_{t=rT+1}^{(r+1)T} Q(t)\left[\delta_t||X_t^* - X_{t-1}||_1 - \nu\right] \\ &- V\sum_{t=rT+1}^{(r+1)T}\hat{\Theta}_t^T X_t^* - \mathcal{D}_r, \end{aligned} \tag{20}$$

where the above inequality follows from the fact that X_t^* is *not* DPP minimizing caching strategy. By adding and subtracting X_{t-1}^* to X_{t-1} in the above, and using triangle inequality, we get the following inequality

$$\begin{aligned} X_T[r] &\leq \sum_{t=rT+1}^{(r+1)T} Q(t)\left[\delta_t||X_t^* - X_{t-1}^*||_1 - \nu\right] - \mathcal{D}_r \\ &- V\sum_{t=rT+1}^{(r+1)T} \hat{\Theta}_t^T X_t^* + \sum_{t=rT+1}^{(r+1)T} Q(t)\delta_t||X_{t-1} - X_{t-1}^*||_1 \\ &\leq \sum_{t=rT+1}^{(r+1)T} Q(rT)\left[\delta_t||X_t^* - X_{t-1}^*||_1 - \nu\right] - \mathcal{D}_r \\ &- V\sum_{t=rT+1}^{(r+1)T} \hat{\Theta}_t^T X_t^* + \sum_{t=rT+1}^{(r+1)T} Q(t)\delta_t||X_{t-1} - X_{t-1}^*||_1 \\ &+ \sum_{t=rT+1}^{(r+1)T} |Q(t) - Q(rT)|\left[\delta_t||X_t^* - X_{t-1}^*||_1 - \nu\right] \end{aligned} \tag{21}$$

where the last inequality above is obtained by adding and subtracting $Q(rT)$, and using the triangle inequality. Using the Lemma 1, we get $|Q(rT) - Q(rT+\tau)| \leq \tau(\delta_{\max}C + \nu)$. Using this in the last term above gives $\sum_{\tau=0}^{T} |Q(t) - Q(rT)|\left[\delta_t||X_t^* - X_{t-1}^*||_1 - \nu\right] \leq \sum_{\tau=0}^{T} \tau(\delta_{\max}C+\nu)^2 \leq \frac{(\delta_{\max}C+\nu)^2T^2}{2}$. Using the fact that $||X_{t-1} - X_{t-1}^*||_1 \leq 2C$, the term $\sum_{t=rT+1}^{(r+1)T} Q(t)\delta_t||X_{t-1} - X_{t-1}^*||_1 \leq 2C\delta_{\max}\sum_{t=rT+1}^{(r+1)T} Q(t)$. Using the Lemma 1, we get

$$\begin{aligned} \sum_{t=rT+1}^{(r+1)T} Q(t) &\leq \sum_{t=rT+1}^{(r+1)T} \left[Q(rT) + 2(t-rT)\delta_{\max}C\right] \\ &\leq TQ(rT) + \frac{\delta_{\max}CT^2}{2}. \end{aligned} \tag{22}$$

Substituting these bounds in (21), we get the following

$$\begin{aligned} X_T[r] &\leq -\mathcal{D}_r - V\sum_{t=rT+1}^{(r+1)T} \hat{\Theta}_t^T X_t^* + C\delta_{\max}TQ(rT) \\ &+ \frac{\delta_{\max}^2C^2T^2 + (\delta_{\max}C+\nu)^2T^2}{2}. \end{aligned} \tag{23}$$

Taking conditional expectation of $X_T[r]$, and using the definition of $\mathcal{D}_r$, it is easy to see that $\mathbb{E}[X_T[r]|\mathcal{F}_{r-1}] \leq 0$; this completes the proof.

10. PROOF OF LEMMA 3

Proving that $Z[R]$ is a Martingale sequence is similar to the proof of Lemma 3 property 1 of [18], and hence omitted. By definition of $X_T[r]$, it is easy to see that

$$|X_T[r]| \leq \delta_{\max}C\sum_{t=rT}^{(r+1)T} Q(t) + \Psi_{T,C}, \tag{24}$$

where $\Psi_{T,C}$ is as defined in the Theorem. In addition, from Lemma 3 in [18], it easily follows that

$$\begin{aligned} |Z[R+1] - Z[R]| &\leq |Y[R+1] - Y[R]| \times \mathbf{1}\{\tau \geq R+1\} \\ &= |X_T[R+1]| \times \mathbf{1}\{\tau \geq R+1\} \\ &\leq \delta_{\max}CTc_1 + \Psi_{T,C}, \end{aligned} \tag{25}$$

where $c_1 = \frac{1}{\alpha}\log\frac{2DR}{\delta}$, and $\Psi_{T,C}$ is as defined in Theorem 1. This completes the proof.

11. PROOF OF THEOREM 2

First, let us state and prove the following Lemma.

Lemma 4 *The caching strategy obtained by the DPP algorithm, i.e., X_t, $t = 0, 1, \ldots$, and the corresponding queue satisfy*

$$\delta_t||X_t - X_{t-1}||_1 \leq \frac{2Vd_{max}C}{Q(t)}. \tag{26}$$

Proof: Unlike the proof of Lemma 5 of [18], here we prove this by using the explicit solution to the DPP problem. This is possible since the problem is a convex program, and the Slater's condition is satisfied. The Lagrangian dual of the DPP problem in Algorithm 1 (line 5) is given by

$$\left[Q(t)\Delta(t) - V\hat{\Theta}_t^T X\right] + \lambda(\mathbf{1}^T X_t - C) - \sum_{i=1}^{F} \beta_i x_{i,t}, \tag{27}$$

where the Lagrangian variables $\lambda \geq 0$ and $\beta_i \geq 0$. Since the problem is convex, at optimal point, by complementary slackness condition, we have $\beta_i^* = 0$. Taking the derivative of the above leads to

$$x_{t,i} - x_{t-1,i} = \frac{V\hat{\theta}_{t,i}}{\delta_t Q(t)}|x_{t,i} - x_{t-1,i}| - \frac{\lambda}{Q(t)\delta_t}|x_{t,i} - x_{t-1,i}|, \tag{28}$$

where the terms involving β_i is ignored as it is zero at optimality. Summing over all i, and using the fact that $\mathbf{1}^T X_t = C$, we get the following solution

$$\lambda^* = \frac{V\sum_i \hat{\theta}_{t,i}|x_{t,i} - x_{t-1,i}|}{\sum_i |x_{t,i} - x_{t-1,i}|}. \tag{29}$$

Substituting the above in (28), and using triangle inequality, we get

$$\begin{aligned} \delta_t||X_t - X_{t-1}||_1 &\leq 2\left|\frac{V\sum_i \hat{\theta}_{t,i}|x_{t,i} - x_{t-1,i}|}{Q(t)}\right| \\ &\leq \frac{2Vd_{\max}C}{Q(t)}. \end{aligned} \tag{30}$$

This completes the proof. Using the definition of the queue update, we have the following bound at time slot τ

$$\begin{aligned} Q(t+\tau-1) &> Q(t+\tau-2) - \nu \\ &> Q(t) - (\tau-1)\nu, \end{aligned} \tag{31}$$

where the last inequality follows from induction. Using this in Lemma 4, we get

$$\delta_{t+\tau-1}||X_{t+\tau-1} - X_{t+\tau-2}||_1 \leq \frac{2Vd_{\max}C}{Q(t) - (\tau-1)\nu}. \quad (32)$$

Choosing the threshold on the queue to be $Q(t) > \frac{4Vd_{\max}CT}{\nu}$ for some $k \in \mathbb{N}$, we have

$$\delta_{t+\tau-1}||X_{t+\tau-1} - X_{t+\tau-2}||_1 \leq \frac{(\tau-1)\nu^2}{4Vd_{\max}CT} \leq \frac{\nu^2}{4Vd_{\max}C}, \quad (33)$$

where the above follows from the fact that $\frac{\tau-1}{T} \leq 1$. The above is less than $1/2$ by choosing $V \geq \frac{\nu^2}{2d_{\max}C}$. The above implies that $\delta_{t+\tau-1}||X_{t+\tau-1} - X_{t+\tau-2}||_1 - \nu < -\frac{T-1}{T}\nu$ for all $\tau = 1, 2, \ldots, T$. Using $t = rT$, and $\tau = T$, the above implies that

$$Q((r+1)T) \leq Q(rT) - (T-1)\nu. \quad (34)$$

The upper bound on the queue drift is straightforward, and hence omitted. This completes the proof of the Lemma.

12. PROOF OF PROPOSITION 1

The proof follows by directly using Lemma 2, and applying Lemma 6 of [18]. The definitions of D, α and ρ also follow directly from Lemma 6 of [18]. From Lemma 6 of [18], the moment bound is given by $\mathbb{E}e^{\alpha Q(rT)} \leq D + (e^{\alpha Q(rT)} - D)\rho^r$, where ρ is as defined in the proposition. The only thing that remains to be proved is $e^{\alpha Q(rT)} - D \leq 0$, which completes the proof. We know that $Q(rT) \leq (\delta_{\max}C + \nu)$. Now consider

$$\begin{aligned} e^{\alpha Q(rT)} - D &\leq e^{\alpha T(\delta_{\max}C+\nu)}\left[1 - \frac{e^{\frac{4\alpha Vd_{\max}CT}{\nu}}}{1-\rho}\right] \\ &+ \frac{\rho}{1-\rho}e^{\frac{4\alpha Vd_{\max}CT}{\nu}} \\ &\leq e^{\frac{4\alpha Vd_{\max}CT}{\nu}}\left[e^{-\alpha T\left(\frac{4VCd_{\max}}{\nu} - (\delta_{\max}C+\nu)\right)} - 1\right] \\ &\leq 0, \end{aligned}$$

where the first inequality follows from the fact that $T \geq 1$, and the last inequality follows since $V \geq \frac{\nu(\delta_{\max}C+\nu)}{4Cd_{\max}}$. This completes the proof.