Competitive Analysis of Online Facility Open Problem

Extended Abstract

Binghan Wu
The University of Sydney
Sydney, Australia
biwu6051@uni.sydney.edu.au

Wei Bao
The University of Sydney
Sydney, Australia
wei.bao@sydney.edu.au

Bing Zhou
The University of Sydney
Sydney, Australia
bing.zhou@sydney.edu.au

ABSTRACT

We investigate an online cost minimization problem of serving requests in a tree of facilities, referred to as the Online Facility Open Problem (Online FOP). To address this problem, we propose the Anchor-Barrier Algorithm (ABA), a threshold-based algorithm applicable to any tree and any cost assignment, which can work in a distributed manner for scalability. We conduct the competitive analysis and show that ABA's achieves the optimal competitive ratio `Height` + 2, where `Height` is the height of the facility tree.

KEYWORDS

Online algorithms; Competitive analysis; Facility location problem

ACM Reference Format:
Binghan Wu, Wei Bao, and Bing Zhou. 2024. Competitive Analysis of Online Facility Open Problem: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

We study an online cost minimization problem for a service system of facilities arranged in a tree topology. All nodes/facilities (used interchangeably) are initially closed and can be opened at any time by paying an opening cost. Each edge has an associated conveyance cost. Requests arrive at leaf nodes sequentially, and all requests must be served, with only open facilities able to do so. When an open facility receives a request, it serves the request without any additional cost. If a closed facility receives a request, it must forward the request to its parent, incurring the associated conveyance cost on the passing edge. If the parent is also closed, the request is recursively forwarded. If the request reaches the root and it is still closed, the request is sent to a faraway always-on facility (outside the tree) at a long-distance conveyance cost. In this paper, we consider a fully heterogeneous setting, where the opening costs/conveyance costs for different facilities/edges can be arbitrary non-negative numbers. The goal is to minimize the total cost without knowledge of whether and where the next request arrives. The challenge lies in deciding when, where and how many facilities to open so as to balance the open costs with the benefits of reduced conveyance costs. We refer to this problem as the Online Facility Open Problem (Online FOP).

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Online FOP is related to, but distinct from, the Facility Location Problem [2, 6, 7, 10–13] and the Multi-Level Aggregation Problem [1, 3, 8] by focusing on conveyance costs within arbitrary tree structures and immediate service requirements respectively. Unlike hierarchical cooperative caching [5, 9, 14, 15], which considers fixed tree depths and distance costs, FOP incorporates "open costs" for facility activation. This unique problem framework allows for the exploration of optimal strategies in tree-structured networks, filling gaps in the current literature and extending applicability to scenarios such as the ski-rental problem with innovative solutions.

In this work, we design the distributed online Anchor-Barrier Algorithm (ABA) in which facilities make decisions based solely on their open costs and the requests they have handled without the need to acquire global information. We consider the online adversarial setting [4], where in the competitive analysis, we fix our algorithm and an adversary deliberately chooses a topology, associated costs and inputs to maximize the competitive ratio.

2 PROBLEM FORMULATION

Our problem is formulated by a rooted tree $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, a conveyance cost vector $\boldsymbol{\lambda}$, an open cost vector $\boldsymbol{\sigma}$, a sequence of requests $\mathcal{R}$, where

- $\mathcal{V} = \{v_1, v_2, v_3, \ldots\}$ is the set of all nodes/facilities considered in the problem.
- W.l.o.g. $v_1 \in \mathcal{V}$ denotes the root of $\mathcal{G}$.
- $\mathcal{E}$ is a set of *undirected* edges.
- $\mathcal{A} \subseteq \mathcal{V}$ is the set of all leaf nodes in $\mathcal{G}$. We use the term leaf nodes and *access facilities* interchangeably in this paper.

Initially, all facilities are off and can be opened by paying a one-shot open cost σ_i, $\forall v_i \in \mathcal{V}$. Let $\mathcal{R}$ denotes the sequence of requests arrive at access facilities. Only open facilities can serve requests, and no extra cost is charged. If a request arrives at a closed facility v_i, v_i can only recursively forward this request to its parent by paying the conveyance cost until an open facility is reached. The conveyance cost of forwarding a request from v_i to its parent is λ_i. Define $\rho(v_i)$ as the parent facility of v_i. Define $\mathcal{P}_{i,k}$ as the sequence of bottom-up connected facilities from v_i to v_k, and define $\mathcal{P}_i$ as the bottom-up sequence from v_i to the root. Define the height of the tree $\mathcal{G}$ as $H_{\mathcal{G}} = \max_i |\mathcal{P}_i| - 1$, $\forall v_i \in \mathcal{A}$, which is the number of edges from the leaf node to the root in the longest path.

Let $OPT(\mathcal{G}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \mathcal{R})$ be the total cost of the optimal offline algorithm that knows all requests $\mathcal{R}$ in advance given $\mathcal{G}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\sigma}$. Let β_i be the indicator of facility v_i being open ($\beta_i = 1$) or closed ($\beta_i = 0$), and let $\alpha_{i,t}$ be the indicator of facility v_i forwarding request r_t ($\alpha_{i,t} = 1$) or not ($\alpha_{i,t} = 0$) for all $v_i \in \mathcal{V}$ and $r_t \in \mathcal{R}$.

Algorithm 1 Anchor-Barrier Algorithm (ABA)

Require: $\mathcal{G}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \mathcal{R}, \boldsymbol{\omega}$
$\theta_n \leftarrow 0, \forall v_n \in \mathcal{V}$ initially
loop
    Upon the arrival of a new request on v_j
    REFERRING(v_j,∞,v_j)
end loop

function REFERRING(v_i, δ, v_j)
    Input: current facility, residual budget, request source
    Output: anchor/open facility, barrier, served
    if v_i is open **then**
        return [v_i, 0, `True`]
    else
        $\delta_{i,j} \leftarrow \min(\delta, \omega_i - \theta_i)$
        if $\delta_{i,j} \geq \lambda_i$ **then** ▷ forward to the parent $\rho(v_i)$
            $v_a, b, s \leftarrow$ REFERRING($\rho(v_i)$,$\delta_{i,j} - \lambda_i$,v_j)
            $\theta_i \leftarrow \theta_i + \lambda_{i,a} + b$
            if $\neg s$ and $\theta_i \geq \omega_i$ **then**
                open v_i
                return [v_a, b, `True`]
            else
                return [v_a, b, s]
            end if
        else ▷ touch barrier, request can not forward
            $\theta_i \leftarrow \theta_i + \delta_{i,j}$ ▷ find barrier $b = \delta_{i,j}$
            if $\theta_i \geq \omega_i$ **then**
                open v_i
                return [v_i, $\delta_{i,j}$, `True`]
            else
                return [v_i, $\delta_{i,j}$, `False`]
            end if
        end if
    end if
end function

The **offline objective** OPT is the solution to the following linear program:

$$\begin{aligned} \min \sum_{i=1}^{|\mathcal{V}|} \beta_i \sigma_i + \sum_{i=1}^{|\mathcal{V}|} \sum_{t=1}^{|\mathcal{R}|} \alpha_{i,t} \lambda_i \\ \text{s.t. } \alpha_{k,t} + \sum_{v_m \in \mathcal{P}_{i,k}} \beta_m \geq 1, \\ \forall r_t \in \mathcal{R}, \ \forall v_k \in \mathcal{P}_i, \text{ where } v_i = r_t \in \mathcal{A}. \end{aligned} \tag{1}$$

Let $ALG(\mathcal{G}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \mathcal{R})$ be the total cost of the **online algorithm** which only knows the request upon its arrival. The **competitive ratio** is defined as the following

$$R^* = \max_{\mathcal{G},\lambda,\sigma,\mathcal{R}} \frac{ALG(\mathcal{G}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \mathcal{R})}{OPT(\mathcal{G}, \boldsymbol{\lambda}, \boldsymbol{\sigma}, \mathcal{R})}. \tag{2}$$

Our goal is to design online algorithms that have the lowest R^*, meaning that the online algorithm is as close to optimal as possible in the worst-case scenario.

3 ALGORITHM

ABA is a threshold based algorithm. In essence, each facility v_i holds an threshold ω_i. Whenever the total cost of forwarding the request θ_i from a facility exceeds its threshold, then we open this facility. Upon the arrival of a request on v_j, our online algorithm starts a distributed referring procedure at the request's arrival facility v_j and recursively sends a reference message to its parent facility. The reference carries the residual budget δ which is the maximum conveyance cost such that the cumulative reference cost of passed facilities does not exceed their threshold. When a facility receives a reference, it decides if it is an anchor facility using the residual budget, then there are 3 possible cases. (1) it is an open facility. This facility is the serving facility for the request, and returns the reference. (2) it is an anchor facility because $\delta_{i,j} < \lambda_i$. Then it computes the location of the barrier, and returns the reference. The facility linking to the barrier will open for this request. (3) otherwise, it forwards the reference to its parent.

4 COMPETITIVE ANALYSIS

The performance of ABA is tied to the threshold. We have the following three Theorems to show the optimal threshold and the performance of ABA. Theorem 1 helps to show the upper bounds of the competitive ratio, and how the competitive ratio degrades as γ changes.

THEOREM 1. *Let $\omega_i = \gamma\sigma_i$, $\forall v_i \in \mathcal{V}$, $\gamma > 0$. The competitive ratio of Algorithm 1 is bounded by $R^* \leq \frac{1}{\gamma}(H_{\mathcal{G}} + 2)$ if $\gamma \leq 1$, otherwise $R^* \leq \gamma(H_{\mathcal{G}} + 2)$.*

Next, Theorem 2 shows that $H_{\mathcal{G}} + 2$ is an optimal and reachable upper bound for a constructive problem instance.

THEOREM 2. *$H_{\mathcal{G}} + 2$ is the optimal tight bound of the competitive ratio, and this bound is uniquely achieved by applying $\gamma = 1$.*

Finally, we have theorem 3 show the optimality of $H_{\mathcal{G}} + 2$.

THEOREM 3. *$H_{\mathcal{G}} + 2$ is the best possible competitive ratio for any deterministic threshold design.*

Given the three theorems, we have proved that ABA is $H_{\mathcal{G}} + 2$-competitive, and this competitive ratio is optimal.

5 CONCLUSION AND FUTURE WORK

In this paper, we study the Online FOP, which aims to minimize total cost by determining where and how much to invest in facilities. We address the Online FOP through ABA, which is a general framework that works on arbitrary tree topologies and cost settings. Moreover, ABA is distributed and scalable, making it suitable for large trees. We perform a competitive analysis and show that ABA is $H_{\mathcal{G}} + 2$ competitive, and this competitive ratio is optimal.

In future work, we propose to extend the scope to a more generalized setting where a currently open facility can be closed and a portion of the open costs can be refunded. Our goal is to capture the concept of divestment of non-performing assets in industrial scenarios. Furthermore, it is also interesting to explore scenarios where references can be shared between siblings, for example in edge collaborative caching, allowing direct communication between edge servers (access facilities).

REFERENCES

[1] Y. Azar and N. Touitou. 2019. General framework for metric optimization problems with delay or with deadlines. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS* 2019-November (2019), 60–71. https://doi.org/10.1109/FOCS.2019.00013

[2] H. Aziz, H. Chan, B.E. Lee, B. Li, and T. Walsh. 2020. Facility location problem with capacity constraints: Algorithmic and mechanism design perspectives. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence* (2020), 1806–1813.

[3] M. Bienkowski, M. Böhm, J. Byrka, M. Chrobak, C. Dürr, L. Folwarczný, Ł. Jez, J. Sgall, N.K. Thang, and P. Veselý. 2016. Online algorithms for multi-level aggregation. *Leibniz International Proceedings in Informatics, LIPIcs* 57 (2016). https://doi.org/10.4230/LIPIcs.ESA.2016.12

[4] Allan Borodin and Ran El-Yaniv. 2005. *Online computation and competitive analysis*. cambridge university press.

[5] X. Chen, Y. Jiang, B. Fan, F.-C. Zheng, D. Niyato, and X. You. 2020. Hierarchical Cooperative Caching in Fog Radio Access Networks: A Brain Storm optimization Approach. *2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings* (2020). https://doi.org/10.1109/GLOBECOM42002.2020.9322173

[6] D. Fotakis. 2003. On the competitive ratio for online facility location. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2719 (2003), 637–652. https://doi.org/10.1007/3-540-45061-0_51

[7] K. Gokbayrak. 2022. A two-level off-grid electric distribution problem on the continuous space. *Computers and Operations Research* 144 (2022). https://doi.org/10.1016/j.cor.2022.105853

[8] Anna R. Karlin, Claire Kenyon, and Dana Randall. 2003. Dynamic TCP acknowledgement and other stories about e/(e - 1). *Algorithmica (New York)* 36, 3 (2003), 209 – 224. https://doi.org/10.1007/s00453-003-1013-x Cited by: 64.

[9] X. Li, C.G. Plaxton, M. Tiwari, and A. Venkataramani. 2006. Online hierarchical cooperative caching. *Theory of Computing Systems* 39, 6 (2006), 851–874. https://doi.org/10.1007/s00224-006-1250-x

[10] C. Markarian. 2021. Online Non-metric Facility Location with Service Installation Costs. *International Conference on Enterprise Information Systems, ICEIS - Proceedings* 1 (2021), 737–743.

[11] David B Shmoys, Éva Tardos, and Karen Aardal. 1997. Approximation algorithms for facility location problems. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. 265–274.

[12] Dong-wan Tcha and Bum-il Lee. 1984. A branch-and-bound algorithm for the multi-level uncapacitated facility location problem. *European Journal of Operational Research* 18, 1 (1984), 35–43.

[13] Tony J Van Roy and Donald Erlenkotter. 1982. A dual-based procedure for dynamic facility location. *Management Science* 28, 10 (1982), 1091–1105.

[14] H. Wu, J. Chen, C. Zhou, W. Shi, N. Cheng, W. Xu, W. Zhuang, and X.S. Shen. 2020. Resource Management in Space-Air-Ground Integrated Vehicular Networks: SDN Control and AI Algorithm Design. *IEEE Wireless Communications* 27, 6 (2020), 52–60. https://doi.org/10.1109/MWC.001.2000130

[15] R. Xie, Q. Tang, and T. Huang. 2019. Energy-efficient hierarchical cooperative caching optimisation for 5G networks. *IET Communications* 13, 6 (2019), 687–695. https://doi.org/10.1049/iet-com.2018.5530