Multi-level Aggregation with Delays and Stochastic Arrivals

Extended Abstract

Mathieu Mari
LIRMM, University of Montpellier
Montpellier, France
mathieu.mari@lirmm.fr

Michał Pawłowski
University of Warsaw, IDEAS NCBR
& Sapienza University of Rome
Warsaw, Poland
michal.pawlowski@mimuw.edu.pl

Runtian Ren
IDEAS NCBR
Warsaw, Poland
runtian.ren@ideas-ncbr.pl

Piotr Sankowski
University of Warsaw, IDEAS NCBR & MIM Solutions
Warsaw, Poland
sank@mimuw.edu.pl

ABSTRACT

In online Multi-Level Aggregation (MLA) with delays, the input is an edge-weighted rooted tree T and a sequence of requests arriving at its vertices (with each vertex representing an independent agent) that need to be served in an online manner. Each request r is characterized by two parameters: its arrival time $t(r)$ and its location $l(r)$ (a vertex). Once r arrives, we can either serve it immediately or postpone this action until any time later. We can serve several pending requests at the same time, paying a service cost equal to the weight of the subtree that contains the locations of all the requests served and the root of T. Postponing the service of a request r to time t generates an additional delay cost of $t - t(r)$. The goal is to serve all requests in an online manner such that the total cost (i.e., the total sum of service and delay costs) is minimized. The MLA problem is a generalization of several well-studied problems, including the TCP Acknowledgment (depth 1), Joint Replenishment (depth 2), and Multi-Level Message Aggregation (arbitrary depth). This problem has only been studied in an adversarial model thus far, and the current best algorithm for this problem achieves a competitive ratio of $O(d^2)$, where d denotes the depth of the tree. We study a stochastic version of MLA where the requests follow a Poisson arrival process. We present a deterministic online algorithm that achieves a constant ratio of expectations, meaning that the ratio between the expected costs of the solution generated by our algorithm and the optimal offline solution is bounded by a constant.

KEYWORDS

Online Algorithms, Multi-level Aggregation, Poisson Arrivals

ACM Reference Format:
Mathieu Mari, Michał Pawłowski, Runtian Ren, and Piotr Sankowski. 2024. Multi-level Aggregation with Delays and Stochastic Arrivals: 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.

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.

1 INTRODUCTION

Imagine the manager of a factory needs to deliver products from the factory to the agents' locations. Once some products are in shortage for some agent, then this agent will inform the factory for replenishment. From the factory's perspective, each time a service is created to deliver the products, a truck has to travel from the factory to go to the locations of the requested agents and then come back to the factory. A cost proportional to the total distance traveled has to be paid for this service. For the purpose of saving delivery costs, it is beneficial to accumulate the replenishment requests from many stores and then deliver the ordered products altogether in one service. However, this accumulated delay in delivering products may cause the agents dissatisfaction, and complaints may negatively influence future contracts between the agents and the factory. Typically, for each request, the time gap between ordering the products and receiving the products is known as the delay cost (of this request). The goal of the factory manager is to plan the delivery schedule in an online manner such that the total service cost and the total delay cost are minimized.

The above is an example of an online optimization problem called Multi-level Aggregation (MLA) with linear delays. Formally, the input is an edge-weighted rooted tree T and a sequence of requests, with each request r specified by an arrival time $t(r)$ and a location at a particular vertex $l(r)$. Once r arrives, its service does not have to be processed immediately but can be delayed to any later time t at a delay cost of $t - t(r)$. The benefit of delaying requests is that several requests can be served together to save some service costs. To serve any set of requests R at time t, a subtree T' containing the tree root and locations of all the requests in R needs to be bought at a service cost equal to the total weight of edges in T'. The goal is to serve all requests in an online manner such that the total cost (i.e., the total service cost plus the total delay cost) is minimized.

The MLA problem has been studied in the adversarial model, and the current best online algorithm achieves a competitive ratio of $O(d^2)$ [12], where d denotes the depth of the tree. The competitive ratio is the ratio between the cost of the online solution and the cost of the optimum offline solution (i.e., knowing in advance all future requests) *for the worst request sequence*. Thus, competitive analysis provides strong bounds on the performance of online algorithms, but worst-case scenarios rarely arise in practice, which makes these results inadequate for understanding real-life scenarios. In fact, it is often too pessimistic to assume that no stochastic information on the input is available in practice — again, consider our delivery example. The factory knows all the historical orders and can estimate the request frequencies from all the stores. Thus, it is reasonable to assume that the requests follow some stochastic distribution. Therefore, the following question is natural: *if stochastic information on the input is available, can we devise online algorithms for MLA with better performance guarantees?* In this paper, we provide an affirmative answer to this question. We study a stochastic online version of MLA, assuming that the requests arrive following a Poisson arrival process [53]. More precisely, the waiting time between any two consecutive requests arriving at the same vertex u follows an exponential distribution $\mathrm{Exp}(\boldsymbol{\lambda}(u))$ with parameter $\boldsymbol{\lambda}(u)$. In this model, the goal is to minimize the expected cost produced by an algorithm ALG for a random input sequence generated in a long time interval $[0, \tau]$. In order to evaluate the performance of an algorithm ALG on stochastic inputs, we use the *ratio of expectations* (RoE), i.e., the ratio between the expected cost of ALG and the expected cost of the optimal offline solution OPT. We prove that the performance guarantee obtained in this model is significantly better compared with the current best competitiveness obtained in the adversarial setting. More specifically, we propose a non-trivial deterministic online algorithm that achieves a constant RoE.

Theorem 1.1. *For MLA with linear delays and Poisson arrivals, there exists a deterministic online algorithm with a constant RoE.*

Previous works. The MLA problem was first introduced by Bienkowski et al. [15] where they study a more general version where the cost of delaying a request r by a duration t is some function $f_r(t)$. They gave an $O(d^4 2^d)$-competitive online algorithm where d denotes the depth of the given tree. This was later improved to $O(d^2)$ [12]. A deadline version of MLA is studied in [15], where each request r has a time window (between its arrival and its deadline), and it has to be served no later than its deadline, and the target is to minimize the total service cost for serving all the requests. For this deadline version, they gave an online algorithm with a competitive ratio $d^2 2^d$, which was later improved to $O(d)$ [24, 50]. The current best lower bound on the competitiveness of MLA with delays is only $2 + \phi \approx 3.618$, restricted to a path case with linear delays [19]. In the offline setting, MLA is NP-hard in both delay and deadline versions [3, 14], and a 2-approximation algorithm is known for the deadline version [14]. When the tree is a path and delay costs are linear functions, the competitiveness is between 3.618 and 5 [19], improving on an earlier 8-competitive algorithm [23]. Thus far, no previous work has studied MLA in the stochastic input model, no matter the delay or deadline versions. Two special cases of MLA with linear delays, one called TCP-acknowledgement ($d = 1$) and one called Joint Replenishment (JRP, $d = 2$), are of particular interest: TCP-acknowledgement (a.k.a. single item lot-sizing problem, [22, 26, 38, 40]) models the data transmission issue from sensor networks [44, 58], while JRP models the inventory control issue from supply chain management [4, 34, 39, 42]. For TCP-acknowledgement, there exists an optimal 2-competitive deterministic algorithm [29] and an optimal $e/(e-1)$-competitive randomized algorithm [41, 54] in the online setting, and it can be solved in polynomial time in the offline setting [1]. For JRP, the competitiveness is between 3 [25] and 2.754 [18]; in the offline setting, JRP is NP-hard [3] and also APX-hard [17, 52]. The current best approximation ratio for JRP is 1.791 [18, 45–47]. For a deadline version of JRP, there exists an optimal 2-competitive algorithm [18]. Recently, many other online problems with delays/deadline have also drawn a lot of attention besides MLA, such as online matching with delays [5, 8, 9, 11, 20, 21, 28, 30, 31, 48, 49, 51], online service with delays [8, 12, 56, 57], facility location with delays/deadline [12, 13, 16], Steiner tree with delays/deadline [13], bin packing with delays [2, 7, 32, 33], set cover with delays [6, 43, 55], paging with delays/deadline [35, 36], list update with delays/deadline [10], and many others [27, 37, 51, 56].

2 THE ALGORITHM: MAIN IDEA

Warm-up: a single edge case. We first consider a single-edge tree case to provide some intuitive ideas. That is, T consists an edge $e = (u, \gamma)$ of weight $w > 0$ and the arrival rate of u is $\lambda > 0$. There exist two opposite strategies for this case. The first strategy, called the *instant strategy*, is to serve each request as soon as it arrives. Intuitively, this approach is efficient when the requests are not so frequent so that, on average, the cost of delaying a request to the arrival time of the next request is enough to compensate for the service cost. The second strategy, called the *periodic approach*, is meant to work in the opposite case where requests are frequent enough so that it is worth grouping several of them for the same service. In this way, the weight cost of a service can be shared between the requests served. Assuming that requests follow some stochastic assumptions, it makes sense to enforce that services are ordered at regular time steps, where the time between any two consecutive services is a fixed number p, which depends only on the instance's parameters. There are two challenges here: (i) when to use each strategy? (ii) what is the value of p that optimizes the performance of the periodic strategy? For the first one, it depends on the value of $\pi := w\lambda$ that we call the *heaviness* of the instance: if $\pi > 1$, i.e., the instance is *heavy*, and the periodic strategy is more efficient; if $\pi \leq 1$, the instance is *light*, and the instant strategy is essentially better. For the second one, the right value for the period, up to a constant in the ratio of expectations, is $p = \sqrt{2w/\lambda}$.

Overview of an online algorithm for a general MLA instance $(T, \boldsymbol{\lambda})$. We generalize the concepts of "light" and "heavy" for trees in a way that the instant and the periodic strategies still essentially work:

- A *light* instance has $\pi(T, \boldsymbol{\lambda}) = \sum_{u \in V(T)} \boldsymbol{\lambda}(u) \cdot d(u, \gamma) \leq 1$, where $d(u, \gamma)$ is the total edges weight on the path from u to γ; for this case, each request is served instantly at its arrival.
- A *heavy* instance has $w_u \geq 1/\boldsymbol{\lambda}(u)$ for all $u \in V(T)$ with $\boldsymbol{\lambda}(u) > 0$, where w_u is the weight of the edge incident to u on the path from u to the root γ; for this case, a period is determined for each u and the requests are served periodically.

Unfortunately, some instances are neither light nor heavy! To deal with such instances, we give an algorithm to partition the tree into two groups of vertices so that the first group essentially corresponds to a light instance (where the instant strategy is applied) while the second group corresponds to a heavy instance (periodic strategy).

ACKNOWLEDGMENTS

This work was partially supported by the ERC CoG grant TUgbOAT no 772346 and NCN grant no 2020/37/B/ST6/04179.

REFERENCES

[1] Alok Aggarwal and James K. Park. 1993. Improved algorithms for economic lot size problems. *Operations research* 41, 3 (1993), 549–571.

[2] Lauri Ahlroth, André Schumacher, and Pekka Orponen. 2013. Online bin packing with delay and holding costs. *Operations Research Letters* 41, 1 (2013), 1–6.

[3] Esther Arkin, Dev Joneja, and Robin Roundy. 1989. Computational complexity of uncapacitated multi-echelon production planning problems. *Operations research letters* 8, 2 (1989), 61–66.

[4] Y. Askoy and S. S. Erenguk. 1988. Multi-item inventory models with coordinated replenishment: a survey. *International Journal of Operations and Production Management* 8 (1988), 63–73.

[5] Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. 2017. Polylogarithmic bounds on the competitiveness of min-cost perfect matching with delays. In *Proc. SODA*. 1051–1061.

[6] Yossi Azar, Ashish Chiplunkar, Shay Kutten, and Noam Touitou. 2020. Set Cover with Delay–Clairvoyance Is Not Required. In *Proc. ESA*. 8:1–8:21.

[7] Yossi Azar, Yuval Emek, Rob van Stee, and Danny Vainstein. 2019. The price of clustering in bin-packing with applications to bin-packing with delays. In *Proc. SPAA*. 1–10.

[8] Yossi Azar, Arun Ganesh, Rong Ge, and Debmalya Panigrahi. 2017. Online service with delay. In *Proc, STOC*. 551–563.

[9] Yossi Azar and Amit Jacob-Fanani. 2020. Deterministic min-cost matching with delays. *Theory of Computing Systems* 64, 4 (2020), 572–592.

[10] Yossi Azar, Shahar Lewkowicz, and Danny Vainstein. 2023. List Update with Delays or Time Windows. *arXiv preprint arXiv:2304.06565* (2023).

[11] Yossi Azar, Runtian Ren, and Danny Vainstein. 2021. The min-cost matching with concave delays problem. In *Proc. SODA*. 301–320.

[12] Yossi Azar and Noam Touitou. 2019. General framework for metric optimization problems with delay or with deadlines. In *Proc. FOCS*. 60–71.

[13] Yossi Azar and Noam Touitou. 2020. Beyond tree embeddings–a deterministic framework for network design with deadlines or delay. In *Proc. FOCS*. 1368–1379.

[14] Luca Becchetti, Alberto Marchetti-Spaccamela, Andrea Vitaletti, Peter Korteweg, Martin Skutella, and Leen Stougie. 2009. Latency-constrained aggregation in sensor networks. *ACM Transactions on Algorithms* 6, 1 (2009), 1–20.

[15] Marcin Bienkowski, Martin Böhm, Jaroslaw Byrka, Marek Chrobak, Christoph Dürr, Lukáš Folwarczný, Łukasz Jeż, Jiří Sgall, Nguyen Kim Thang, and Pavel Veselý. 2016. Online Algorithms for Multi-Level Aggregation. In *Proc. ESA*. 12:1–12:17.

[16] Marcin Bienkowski, Martin Böhm, Jarosław Byrka, and Jan Marcinkowski. 2022. Online Facility Location with Linear Delay. In *Proc. APPROX/RANDOM*. 45:1–45:17.

[17] Marcin Bienkowski, Jarosław Byrka, Marek Chrobak, Neil Dobbs, Tomasz Nowicki, Maxim Sviridenko, Grzegorz Świrszcz, and Neal E. Young. 2015. Approximation algorithms for the joint replenishment problem with deadlines. *Journal of Scheduling* 18, 6 (2015), 545–560.

[18] Marcin Bienkowski, Jaroslaw Byrka, Marek Chrobak, Łukasz Jeż, Dorian Nogneng, and Jiří Sgall. 2014. Better approximation bounds for the joint replenishment problem. In *Proc. SODA*. 42–54.

[19] Marcin Bienkowski, Jaroslaw Byrka, Marek Chrobak, Łukasz Jeż, Jiří Sgall, and Grzegorz Stachowiak. 2013. Online control message aggregation in chain networks. In *Proc. WADS*. 133–145.

[20] Marcin Bienkowski, Artur Kraska, Hsiang-Hsuan Liu, and Paweł Schmidt. 2018. A primal-dual online deterministic algorithm for matching with delays. In *Proc. WAOA*. 51–68.

[21] Marcin Bienkowski, Artur Kraska, and Paweł Schmidt. 2017. A match in time saves nine: Deterministic online matching with delays. In *Proc. WAOA*. 132–146.

[22] Nadjib Brahimi, Stéphane Dauzere-Peres, Najib M Najid, and Atle Nordli. 2006. Single item lot sizing problems. *European Journal of Operational Research* 168, 1 (2006), 1–16.

[23] Carlos Fisch Brito, Elias Koutsoupias, and Shailesh Vaya. 2012. Competitive analysis of organization networks or multicast acknowledgment: How much to wait? *Algorithmica* 64 (2012), 584–605.

[24] Niv Buchbinder, Moran Feldman, Joseph Naor, and Ohad Talmon. 2017. O (depth)-competitive algorithm for online multi-level aggregation. In *Proc. SODA*. 1235–1244.

[25] Niv Buchbinder, Tracy Kimbrelt, Retsef Levi, Konstantin Makarychev, and Maxim Sviridenko. 2008. Online make-to-order joint replenishment model: primal dual competitive algorithms. In *Proc. SODA*. 952–961.

[26] Maxim A. Bushuev, Alfred Guiffrida, M.Y. Jaber, and Mehmood Khan. 2015. A review of inventory lot sizing review papers. *Management Research Review* 38, 3 (2015), 283–298.

[27] Ryder Chen, Jahanvi Khatkar, and Seeun William Umboh. 2022. Online Weighted Cardinality Joint Replenishment Problem with Delay. In *Proc. ICALP*.

[28] Lindsey Deryckere and Seeun William Umboh. 2022. Online Matching with Set Delay. *arXiv preprint arXiv:2211.02394* (2022).

[29] Daniel R. Dooly, Sally A. Goldman, and Stephen D. Scott. 2001. On-line analysis of the TCP acknowledgment delay problem. *J. ACM* 48, 2 (2001), 243–273.

[30] Yuval Emek, Shay Kutten, and Roger Wattenhofer. 2016. Online matching: haste makes waste!. In *Proc. STOC*. 333–344.

[31] Yuval Emek, Yaacov Shapiro, and Yuyi Wang. 2019. Minimum cost perfect matching with delays for two sources. *Theoretical Computer Science* 754 (2019), 122–129.

[32] Leah Epstein. 2021. On bin packing with clustering and bin packing with delays. *Discrete Optimization* 41 (2021), 100647.

[33] Leah Epstein. 2022. Open-end bin packing: new and old analysis approaches. *Discrete Applied Mathematics* 321 (2022), 220–239.

[34] Suresh K. Goyal and Ahmet T. Satir. 1989. Joint replenishment inventory control: deterministic and stochastic models. *European journal of operational research* 38, 1 (1989), 2–13.

[35] Anupam Gupta, Amit Kumar, and Debmalya Panigrahi. 2020. Caching with time windows. In *Proc. STOC*. 1125–1138.

[36] Anupam Gupta, Amit Kumar, and Debmalya Panigrahi. 2022. A Hitting Set Relaxation for k-Server and an Extension to Time-Windows. In *Proc. FOCS*. 504–515.

[37] Sungjin Im, Benjamin Moseley, Chenyang Xu, and Ruilong Zhang. 2023. Online Dynamic Acknowledgement with Learned Predictions. *arXiv preprint arXiv:2305.18227* (2023).

[38] Raf Jans and Zeger Degraeve. 2008. Modeling industrial lot sizing problems: a review. *International Journal of Production Research* 46, 6 (2008), 1619–1643.

[39] Dev Joneja. 1990. The joint replenishment problem: new heuristics and worst case performance bounds. *Operations Research* 38, 4 (1990), 711–723.

[40] Behrooz Karimi, SMT Fatemi Ghomi, and JM Wilson. 2003. The capacitated lot sizing problem: a review of models and algorithms. *Omega* 31, 5 (2003), 365–378.

[41] Anna R. Karlin, Claire Kenyon, and Dana Randall. 2001. Dynamic TCP acknowledgement and other stories about e/(e-1). In *Proc. STOC*. 502–509.

[42] Moutaz Khouja and Suresh Goyal. 2008. A review of the joint replenishment problem literature: 1989–2005. *European journal of operational Research* 186, 1 (2008), 1–16.

[43] Ngoc Mai Le, Seeun William Umboh, and Ningyuan Xie. 2023. The Power of Clairvoyance for Multi-Level Aggregation and Set Cover with Delay. In *Proc. SODA*. 1594–1610.

[44] Ka-Cheong Leung, Victor OK Li, and Daiqin Yang. 2007. An overview of packet reordering in transmission control protocol (TCP): problems, solutions, and challenges. *IEEE Transactions on Parallel and Distributed Systems* 18, 4 (2007), 522–535.

[45] Retsef Levi, Robin Roundy, David Shmoys, and Maxim Sviridenko. 2008. A constant approximation algorithm for the one-warehouse multiretailer problem. *Management Science* 54, 4 (2008), 763–776.

[46] Retsef Levi, Robin Roundy, and David B Shmoys. 2004. Primal-dual algorithms for deterministic inventory problems. In *Proc. STOC*. 353–362.

[47] Retsef Levi and Maxim Sviridenko. 2006. Improved approximation algorithm for the one-warehouse multi-retailer problem. In *Proc. APPROX-RANDOM*. 188–199.

[48] Xingwu Liu, Zhida Pan, Yuyi Wang, and Roger Wattenhofer. 2018. Impatient online matching. In *Proc. ISAAC*, Vol. 123. 62:1–62:12.

[49] Mathieu Mari, Michał Pawłowski, Runtian Ren, and Piotr Sankowski. 2023. Online matching with delays and stochastic arrival times. In *Proc. AAMAS*. 976–984.

[50] Jeremy McMahan. 2021. A D-competitive algorithm for the Multilevel Aggregation Problem with Deadlines. *arXiv preprint arXiv:2108.04422* (2021).

[51] Darya Melnyk, Yuyi Wang, and Roger Wattenhofer. 2021. Online k-Way Matching with Delays and the H-Metric. *arXiv preprint arXiv:2109.06640* (2021).

[52] Tim Nonner and Alexander Souza. 2009. Approximating the joint replenishment problem with deadlines. *Discrete Mathematics, Algorithms and Applications* 1, 02 (2009), 153–173.

[53] Sheldon M. Ross. 1996. *Stochastic processes*. Vol. 2. Wiley New York.

[54] Steven S. Seiden. 2000. A guessing game and randomized online algorithms. In *Proc. STOC*. 592–601.

[55] Noam Touitou. 2021. Nearly-Tight Lower Bounds for Set Cover and Network Design with Deadlines/Delay. In *Proc. ISAAC*. 53:1–53:16.

[56] Noam Touitou. 2023. Frameworks for Nonclairvoyant Network Design with Deadlines or Delay. In *Proc. ICALP*. 105:1–105:20.

[57] Noam Touitou. 2023. Improved and Deterministic Online Service with Deadlines or Delay. In *Proc. STOC*. 761–774.

[58] Wei Yuan, Srikanth V. Krishnamurthy, and Satish K. Tripathi. 2003. Synchronization of multiple levels of data fusion in wireless sensor networks. In *Proc. GLOBECOM*, Vol. 1. 221–225.