Online Matching: Haste makes Waste!
(Full Version)[*]

Yuval Emek[†]   Shay Kutten[‡]   Roger Wattenhofer[§]

Abstract

This paper studies a new online problem, referred to as *min-cost perfect matching with delays (MPMD)*, defined over a finite metric space (i.e., a complete graph with positive edge weights obeying the triangle inequality) $\mathcal{M}$ that is known to the algorithm in advance. Requests arrive in a continuous time online fashion at the points of $\mathcal{M}$ and should be served by matching them to each other. The algorithm is allowed to delay its request matching commitments, but this does not come for free: the total cost of the algorithm is the sum of metric distances between matched requests *plus* the sum of times each request waited since it arrived until it was matched. A randomized online MPMD algorithm is presented whose competitive ratio is $O(\log^2 n + \log \Delta)$, where n is the number of points in $\mathcal{M}$ and Δ is its aspect ratio. The analysis is based on a machinery developed in the context of a new stochastic process that can be viewed as two interleaved Poisson processes; surprisingly, this new process captures precisely the behavior of our algorithm. A related problem in which the algorithm is allowed to clear any unmatched request at a fixed penalty is also addressed. It is suggested that the MPMD problem is merely the tip of the iceberg for a general framework of online problems with delayed service that captures many more natural problems.

1   Introduction

Consider an online gaming platform supporting two-player games such as Chess, Scrabble, or Street Fighter 4.[1] The platform tries to find a suitable opponent for each player connecting to it; matching two players initiates a new game between them. The platform should minimize two criteria: (i) the difference between the matched players' rating (a positive integer that represents the player's skill), so that the game is challenging for both players; and (ii) the waiting time until a player is matched and can start playing since waiting is boring. (In reality, the 1-dimensional player rating space is often generalized to a more complex metric space by taking into account additional parameters such as the network distance between the matched players.) It turns out, though, that these two minimization criteria are often conflicting: What if the pool of players waiting for a suitable opponent does not contain anyone whose rating is close to that of a new player? Should the system match the new player to an opponent whose rating differs significantly from hers?

The naive approach that matches players immediately does a terrible job: Murphy's Law may strike, and right after matching a player, a perfect opponent will emerge: Haste makes waste, unbounded waste in fact. To cope with this challenge, we must allow the platform to delay its service in a *rent-or-buy* manner.

Model.   Let $\mathcal{M} = (V, \delta)$ be a finite metric space. Consider a set R of *requests*, where each request $\rho \in R$ is characterized by its *location* $\ell(\rho) \in V$ (also referred to as the point that *hosts* ρ) and *arrival time* $t(\rho) \in \mathbb{R}_{\geq 0}$.[2] Assume for the time being that $|R|$ is even.[3] Notice that R can have multiple requests with the same location (in particular, $|R|$ is unbounded with respect to $|V|$); for simplicity, we assume that each request has a unique arrival time.[4]

The input to an online algorithm for the *min-cost perfect matching with delays (MPMD)* problem is a finite metric space $\mathcal{M}$, provided to the algorithm before the execution commences, and a request set R over $\mathcal{M}$ such that each request $\rho \in R$ is presented to the algorithm in an online fashion at its arrival time $t(\rho)$. The goal of the algorithm is to construct a (perfect) *matching* of the request set — namely, a partition of R into $|R|/2$ unordered request pairs — in an online fashion with no preemption.

The algorithm is allowed to delay the matching of any request in R at a cost. More formally, the requests ρ_1 and ρ_2 can be matched at any time $t \geq \max\{t(\rho_1), t(\rho_2)\}$; if algorithm $\mathcal{A}$ matches requests ρ_1 and ρ_2 at time t, then it incurs a *time cost* of $\mathrm{cost}_{\mathcal{A}}^t(\rho_i) = t - t(\rho_i)$ and a *space cost* of $\mathrm{cost}_{\mathcal{A}}^s(\rho_i) = \delta(\ell(\rho_1), \ell(\rho_2))/2$ for serving ρ_i, $i \in \{1, 2\}$. The *space cost* and *time cost* of $\mathcal{A}$

---

[1]Leading gaming platforms include XBOX Live and the Playstation Network for consoles, Steam for PCs, and web-based platforms such as Geewa, Pogo and Yahoo Games.

[2] For ease of reference, Tab. 1 provides an index for the notation used throughout this paper.

[3]The problem presented here is not well defined if $|R|$ is odd however, later on we discuss a variant of this problem that is well defined for any finite $|R|$.

[4]This assumption is without loss of generality as the arrival times can be slightly perturbed.

for the whole request set are $\mathrm{cost}^s_{\mathcal{A}}(R, \mathcal{M}) = \sum_{\rho \in R} \mathrm{cost}^s_{\mathcal{A}}(\rho)$ and $\mathrm{cost}^t_{\mathcal{A}}(R, \mathcal{M}) = \sum_{\rho \in R} \mathrm{cost}^t_{\mathcal{A}}(\rho)$, respectively. The objective is to minimize $\mathrm{cost}_{\mathcal{A}}(R, \mathcal{M}) = \mathrm{cost}^s_{\mathcal{A}}(R, \mathcal{M}) + \mathrm{cost}^t_{\mathcal{A}}(R, \mathcal{M})$. When $\mathcal{A}$ is clear from the context, we may drop the subscript.

Following the common practice in online computation (cf. [12]), the quality of an online MPMD algorithm is measured in terms of its *competitive ratio*. Online MPMD algorithm $\mathcal{A}$ is said to be α-*competitive* if for every finite metric space $\mathcal{M}$, there exists some $\beta = \beta(\mathcal{M})$ such that for every (even size) request set R over $\mathcal{M}$, it is guaranteed that $\mathbb{E}[\mathrm{cost}_{\mathcal{A}}(R, \mathcal{M})] \leq \alpha \cdot \mathrm{cost}_{\mathcal{A}^*}(R, \mathcal{M}) + \beta$, where the expectation is taken over the coin tosses of the algorithm (if any) and $\mathcal{A}^*$ is an optimal offline algorithm. It is assumed that $\mathcal{M}$ and R are generated by an *oblivious adversary* that knows $\mathcal{A}$, but not the realization of its coin tosses.

Related work. The *rent-or-buy* feature is fundamental to many online applications and thus, prominent in the theoretical study of online computation. Classic online problems in which the rent-or-buy feature constitutes the sole source of difficulty include ski-rental [27, 26, 25] and TCP acknowledgment [16, 17, 25]. In other problems, the rent-or-buy feature is combined with a complex combinatorial structure, enhancing an already challenging online problem, e.g., the extension of online job scheduling [5, 4] studied in [3, 2, 6].

The *matching* problem is a combinatorial optimization celebrity ever since the seminal work of Edmonds [19, 18]. The realm of *online algorithms* also features an extensive literature on matching and some generalizations thereof. Online problems that have been studied in this regard include maximum cardinality matching [28, 11, 22, 15, 33, 34], maximum vertex-weighted matching [1, 15, 34], maximum capacitated assignment (a.k.a. the AdWords problem) [31, 13, 22, 1, 34], metric maximum weight matching [23, 29], metric minimum cost perfect matching [23, 32, 7], and metric minimum capacitated assignment (a.k.a. the transportation problem) [24]; see [30] for a comprehensive survey. All these online problems are *bipartite* matching versions, where the nodes in one side of the graph are static and the nodes in the other side are revealed in an online fashion together with their incident edges.

Discussion and results. To the best of our knowledge, the MPMD problem is the first online *all-pairs* matching version. Moreover, in contrast to the previously studied online matching versions, in MPMD the graph (or metric space) is known a-priori and the algorithmic challenge stems from the unknown locations and arrival times of the requests (whose number is unbounded); this is more in the spirit of online problems such as the classic k-server problem.

The main technical result of this paper is a randomized online MPMD algorithm whose competitive ratio is $O(\log^2 n + \log \Delta)$, where n is the number of points in the metric space $\mathcal{M}$ and Δ is its aspect ratio. This algorithm, presented in Sec. 3, is based on *exponential timers* that determine how long should we wait before committing to a certain match. The analysis, presented in Sec. 4,

relies heavily on machinery we develop in the context of a new stochastic process named *alternating Poisson process.*

We also consider a variant of the online MPMD problem, referred to as *MPMDfp*, in which the algorithm can clear any unmatched request at a fixed penalty. This problem variant is motivated by noticing that clearing an unmatched request may correspond to matching a player with a computer opponent in the context of the aforementioned gaming platforms. The MPMDfp problem is discussed further in Sec. 5, where we show that our online algorithm can be adjusted to cope with this variant as well.

It is not difficult to develop constant lower bounds on the competitive ratio of online MPMD algorithms already for the special case of a 2-point metric space (note that this special case generalizes the ski rental problem). While a 2-point metric space admits an $O(1)$-competitive online MPMD algorithm, we conjecture that in the general case, the competitive ratio must grow as a function of n. In particular, we believe that this conjecture holds for the 1-dimensional metric spaces constructed in Appendix C of [20] (a variant of the construction in Fig. 1 of [36]). We also establish an algorithm-specific lower bound: To demonstrate the role of randomness in the online algorithm presented in Sec. 3, we show in Sec. 6 that the competitive ratio of its natural deterministic counterpart is $\Omega(n)$.

Online problems with delayed service. The online MPMD problem is obtained by augmenting the (offline) min-cost perfect matching problem with the time axis over which service can be delayed in a rent-or-buy manner. This viewpoint seems to open a gate to a general framework of online problems with delayed service since the approach of combining the rent-or-buy feature with a combinatorial optimization offline problem can be applied to a class of minimization problems much larger than just min-cost perfect matching.

To be more precise, consider a minimization problem $\mathcal{P}$ defined with respect to some underlying combinatorial structure $\mathcal{C}$ with a ground set $\mathcal{E}_{\text{in}}$ of *input entities* and a ground set $\mathcal{E}_{\text{out}}$ of *output entities*. The input and output instances of $\mathcal{P}$ are multisets over $\mathcal{E}_{\text{in}}$ and $\mathcal{E}_{\text{out}}$, respectively. For each input instance I, problem $\mathcal{P}$ determines a collection $\mathcal{F}(I)$ of *feasible* output instances; input instance I is said to be *admissible* if $|\mathcal{F}(I)| \neq \emptyset$. We restrict our attention to problems $\mathcal{P}$ satisfying the property that for every two input instances $I \subseteq J$, if I and J are admissible, then so is $J - I$.[5]

Minimization problem Π can be transformed into an online problem with delayed service Π_{on} by applying to it the *delayed service operator*: Each request in Π_{on} is characterized by its *location* — an entity in $\mathcal{E}_{\text{in}}$ — and by its arrival time. The algorithm can serve a collection R of yet unserved requests by buying a feasible (under $\mathcal{F}$) output instance S for their location multiset at any time

---

[5]We follow the standard multiset convention that for two multisets M, N over a ground set S with multiplicity functions $\mu_M : S \to \mathbb{Z}_{\geq 0}$ and $\mu_N : S \to \mathbb{Z}_{\geq 0}$, the relation $M \subseteq N$ holds if $\mu_M(x) \leq \mu_N(x)$ for every $x \in S$; and $N - M$ is the multiset whose multiplicity function $\mu_{N-M} : S \to \mathbb{Z}_{\geq 0}$ satisfies $\mu_{N-M}(x) = \max\{\mu_N(x) - \mu_M(x), 0\}$ for every $x \in S$.

t after the arrival of all requests in R. The payment for this service is the cost of S plus the total waiting times of the requests in R up to time t. Notice that this act of buying S does not serve requests other than those in R including any request arriving at the locations of R after time t.

The online MPMD problem is obtained by applying this delayed service operator to the metric min-cost perfect matching problem, where $\mathcal{C}$ is a finite metric space, $\mathcal{C}_{\text{in}}$ is its points, and $\mathcal{C}_{\text{out}}$ is the set of unordered point pairs (a point multiset is an admissible input instance if its cardinality is even).[6] This operator can also be applied to the vertex cover problem ($\mathcal{C}$ is a graph, $\mathcal{C}_{\text{in}}$ is the edge set, and $\mathcal{C}_{\text{out}}$ is the vertex set), the dominating set problem ($\mathcal{C}$ is a graph and $\mathcal{C}_{\text{in}}$ and $\mathcal{C}_{\text{out}}$ are the vertex set), and many more combinatorial optimization problems.

2 Preliminaries

Tree notation and terminology. Consider a tree T *rooted* at some vertex r with a *leaf* set $\mathcal{L}$. The notions *parent*, *ancestor*, *child*, and *sibling* are used in their usual sense. A *binary* tree is called *full* if every internal vertex has exactly two children.

Let v be some vertex in T. The parent of v in T (assuming that $v \neq r$) is denoted by $p(v)$. We denote the *subtree* of T rooted at v by $T(v)$ and the leaf set of $T(v)$ by $\mathcal{L}(v)$. The set of ancestors of v (excluding v itself) is denoted by $anc(v)$. The *depth* of v in T — i.e., the distance (in hops) from v to r — is denoted by $\text{depth}(v)$ and the *height* of T is denoted by $\text{height}(T) = \max_{x \in \mathcal{L}} \text{depth}(x)$.

A *stilt* in T is an oriented path connecting some vertex $v \in T$, referred to as the *head* of the stilt, with a leaf in $\mathcal{L}(v)$, referred to as the *foot* of the stilt. Given two leaves $x, y \in \mathcal{L}$, their *least common ancestor (LCA)* in T is denoted by $\text{lca}(x, y)$.

Probabilistic embedding in tree metric spaces. Let $w : T \to \mathbb{R}_{\geq 0}$ be a weight function on the vertices of T that satisfies (i) $w(v) = 0$ for every leaf $v \in \mathcal{L}$; and (ii) $w(v) < w(p(v))$ for every vertex $v \in T - \{r\}$. The pair (T, w) introduces a finite metric (in fact, an ultrametric) space over the leaf set $\mathcal{L}$ with distance function δ defined by setting $\delta(x, y) = w(\text{lca}(x, y))$ for every $x, y \in \mathcal{L}$. A metric space that can be realized by such a (T, w) pair is referred to as a *tree metric space*. We subsequently identify a tree metric space with the pair (T, w) that realizes it.

Consider some real $\alpha > 1$. A *hierarchically well separated tree* with parameter α (cf. [8]), or *α-HST* in short, is a tree metric space (T, w) that, in addition to the aforementioned requirements, satisfies $w(p(v)) \geq \alpha \cdot w(v)$ for every vertex $v \in T - \{r\}$. We refer to an α-HST realized by a full binary tree T (cf. [14]) as an *α-HSBT*.

---

[6]In the offline version of the metric min-cost perfect matching problem it suffices to consider only sets (rather than multisets) for the input and output instances. The generalization to multisets is necessary for the transition to the online version of the problem.

The following theorem is established by combining a celebrated construction of Fakcharoenphol et al. [21] (improving previous constructions of Bartal [8, 9]; see also [10]) with a tree transformation technique [35] (details are deferred to Apx. A).

Theorem 2.1. *Consider some n-point metric space (V, δ) of aspect ratio $\Delta = \frac{\max_{x \neq y \in V} \delta(x,y)}{\min_{x \neq y \in V} \delta(x,y)}$ and let $\mathcal{U}$ be the set of all $(1 + \Omega(1/\log n))$-HSBTs (T, w) over V with $\mathrm{height}(T) = O(\log \Delta + \log n)$ and with distance functions $\delta_{\mathcal{T}}$ that dominate δ in the sense that $\delta_{\mathcal{T}}(x, y) \geq \delta(x, y)$ for every $x, y \in V$. There exists a probability distribution $\mathcal{P}$ over $\mathcal{U}$ such that $\mathbb{E}_{(V, \delta_{\mathcal{T}}) \in \mathcal{P}}[\delta_{\mathcal{T}}(x, y)] \leq O(\log n) \cdot \delta(x, y)$ for every $x, y \in V$. Moreover, the probability distribution $\mathcal{P}$ can be sampled efficiently.*

Matching algorithm notation and terminology. Consider the operation of an MPMD algorithm on some HSBT (T, w). Recall that the input to the algorithm consists of a finite set R of requests, where each request $\rho \in R$ is characterized by its location $\ell(\rho) \in \mathcal{L}$ and arrival time $t(\rho) \in \mathbb{R}_{\geq 0}$. Suppose that the algorithm matches requests ρ and ρ' with $\ell(\rho) = x \in \mathcal{L}$ and $\ell(\rho') = x' \in \mathcal{L}$, $x \neq x'$. Let v be some vertex in the unique path connecting x and x' in T. If $v = \mathrm{lca}(x, x')$, then we refer to this matching operation as matching *across* v; otherwise, we refer to it as matching *on top of* v. Notice that matching across v corresponds to matching a request located in $\mathcal{L}(u_1)$ with a request located in $\mathcal{L}(u_2)$, where u_1 and u_2 are the children of v in T, whereas matching on top of v corresponds to matching a request located in $\mathcal{L}(v)$ with a request located in $\mathcal{L} - \mathcal{L}(v)$.

If the algorithm matches request $\rho \in R$ at time t', then ρ is said to be *active* at all times $t(\rho) \leq t < t'$. Given some vertex $v \in T$, we denote the set of active requests in $\mathcal{L}(v)$ at time t by $C_v(t)$ and write $C(t) = C_r(t)$. Vertex v is said to be *odd* at time t if $|C_v(t)| = 1 \pmod 2$; let $D(t)$ be the set of odd vertices at time t.

A key observation is that the forest induced on T by the vertex subset $D(t)$ is a collection — denoted hereafter by $\mathcal{S}(t)$ — of vertex disjoint stilts. Moreover, if v is the head of a stilt in $\mathcal{S}(t)$ then either (1) $v = r$ is the root of T (which implies that $|C(t)|$ is odd); or (2) the sibling of v is also the head of a stilt in $\mathcal{S}(t)$. Let $H(t) \subseteq D(t)$ be the set of heads of stilts in $\mathcal{S}(t)$.

Internal vertex $v \in T - \mathcal{L}$ is said to be *effective* at time t if its two children are odd (which, in particular, means that v is not odd); let $F(t)$ be the set of effective vertices at time t. Notice that v is effective if and only if its two children are in $H(t)$ and let $S_1, S_2 \in \mathcal{S}(t)$ be their corresponding stilts. We refer to the feet of S_1 and S_2 as the *supporting leaves* of v at time t.

We shall apply the aforementioned matching algorithm definitions to both our online MPMD algorithm, denoted by $\mathcal{A}$, and to the benchmark offline MPMD algorithm, denoted by $\mathcal{A}^*$. To distinguish between the two, we reserve the aforementioned notation system for the former and add a superscript asterisk for the latter; in particular, the set of vertices odd under $\mathcal{A}^*$ at time t is denoted by $D^*(t)$ (whereas the set of vertices odd under $\mathcal{A}$ at time t is denoted by $D(t)$).

3 An online MPMD algorithm

In this section, we present our online MPMD algorithm, referred to as the *stilt-walker algorithm* and denoted hereafter by $\mathcal{A}$; its competitive ratio is analyzed in Sec. 4. The algorithm works in two stages: a preprocessing stage, in which we employ Thm. 2.1 to embed the input metric space in a random $(1 + \Omega(1/\log n))$-HSBT (T, w), and the actual online execution, in which $\mathcal{A}$ processes the requests arriving at the leaves of T and constructs the desired matching. The remainder of this section is dedicated to describing the latter.

The matching policy. Although $\mathcal{A}$ operates in continuous time, it will be convenient to describe it as if it progresses in discrete *time steps*, taking the difference dt between two consecutive time steps to be infinitesimally small so that at most one request arrives in each time step.

Fix some time step t. If request ρ arrives at this time step and $\ell(\rho)$ already hosts another active (under $\mathcal{A}$) request ρ', then the algorithm matches ρ and ρ' immediately. Assume hereafter that each leaf in $\mathcal{L}$ hosts at most one active request.

Consider some effective vertex $v \in F(t)$ and let x_1^v, x_2^v be its supporting leaves (the feet of the corresponding stilts in $\mathcal{S}(t)$). By definition, x_i^v hosts an odd number of active requests at time t for $i \in \{1, 2\}$ and since it cannot host more than one active request, it follows that there exists a unique active request ρ_i^v at time t with $\ell(\rho_i^v) = x_i^v$; we refer to ρ_1^v and ρ_2^v as the *supporting requests* of v. The algorithm tosses an independent biased coin and matches its supporting requests (i.e., matching across v) with probability $dt/w(v)$. In what follows, we attribute this coin toss to v so that we can distinguish between coin tosses of different (internal) vertices. A pseudocode description of the stilt-walker algorithm is provided in Pseudocode 1.

Pseudocode 1 The operation of $\mathcal{A}$ at time step t.

1: **if** $\exists \rho, \rho' \in C(t)$ with $\ell(\rho) = \ell(\rho')$ **then**   ▷ there can be at most one such request pair
2:     match ρ and ρ'
3: **end if**
4: **for all** $v \in F(t)$ **do**
5:     $x_1^v, x_2^v \leftarrow$ supporting leaves of v
6:     $\rho_i^v \leftarrow$ unique active request with $\ell(\rho_i^v) = x_i^v$ for $i = 1, 2$
7:     $z = z(v, t) \leftarrow$ outcome of an independent Bernoulli trial with parameter $dt/w(v)$
8:     **if** $z = 1$ **then**
9:         match ρ_1^v and ρ_2^v ▷ matching across v
10:     **end if**
11: **end for**

An analogous "continuous" description of the stilt-walker algorithm's policy regarding the effective vertices is as follows. Consider some internal vertex $v \in T - \mathcal{L}$ and suppose that the last

time $\mathcal{A}$ matched across v was at time t_0 (take $t_0 = 0$ if $\mathcal{A}$ still has not matched across v). Then, the next time the algorithm matches across v is the minimum t_1 that satisfies

$$\int_{t_0}^{t_1} \mathbf{1}(v \in F(t)) \, dt = Z \, ,$$

where $\mathbf{1}(\cdot)$ denotes the indicator operator and $Z = Z(v, t_0) \sim \mathrm{Exp}(1/w(v))$ is an (independent) random variable that obeys an exponential distribution with rate $1/w(v)$.

> **Intuition spotlight:** *The reader may wonder about the role of the exponential timers main-*
> *tained at the internal vertices. At first, we tried to analyze the deterministic version of the*
> *algorithm, where the $(1/w(v))$-rate exponential timer maintained at vertex $v \in T - \mathcal{L}$ is replaced*
> *by a deterministic $\Theta(w(v))$-timer. This seemed to make sense because it allows the algorithm*
> *to wait for $\Theta(w(v))$ time before it pays $w(v)$ in space cost (the usual approach to rent-or-buy*
> *problems). However, as demonstrated in Sec. 6, this is hopeless. Switching to the randomized*
> *version resolves this obstacle because the memoryless exponential timers allow us to analyze*
> *each vertex independently and partition the time into periods so that each period can be ana-*
> *lyzed independently — see Sec. 4.2.*

Notice that our algorithm is guaranteed to eventually match all requests with probability 1. Indeed, if there are at least two active requests at time t, then there is at least one effective vertex v at time t and $\mathcal{A}$ matches across it (thus matching its supporting requests) at time $t + dt$ with probability $dt/w(v)$.

4   Analyzing the stilt-walker algorithm

Our main goal in this section is to establish the following Theorem.

Theorem 4.1. *Fix some $1 < \alpha \leq 2$ and consider an α-HSBT $\mathcal{T}$ realized by a full binary tree of height h. Let R be a request set over $\mathcal{T}$ and let $\mathcal{A}^*$ be some benchmark offline MPMD algorithm for $\mathcal{T}$, R. The stilt-walker algorithm $\mathcal{A}$ guarantees that*

$$\mathbb{E}\left[\mathrm{cost}_{\mathcal{A}}(R, \mathcal{T})\right] \leq O(1/(\alpha-1)) \cdot \mathrm{cost}^s_{\mathcal{A}^*}(R, \mathcal{T}) + O(h) \cdot \mathrm{cost}^t_{\mathcal{A}^*}(R, \mathcal{T}) + \beta \, ,$$

where $\beta = \beta(\mathcal{T})$ depends only on $\mathcal{T}$ and is independent of R.

We will soon turn our attention to the proof of Thm. 4.1, but first, let us show that it yields the desired upper bound on the competitive ratio of $\mathcal{A}$. To that end, fix some n-point metric space $\mathcal{M} = (V, \delta)$ of aspect ratio Δ and a request set R over $\mathcal{M}$ and let $\widetilde{\mathcal{A}}^*$ be an optimal (offline) algorithm for R (over $\mathcal{M}$). Let $\mathcal{P}$ be the probability distribution promised by Thm. 2.1 when applied to $\mathcal{M}$. Denoting the coin tosses of $\mathcal{A}$ by χ and taking $\mathcal{T}$ to be some HSBT in the support of $\mathcal{P}$, we can employ Thm. 4.1 to conclude that

$$\mathbb{E}_{\chi}\left[\mathrm{cost}_{\mathcal{A}}(R, \mathcal{T})\right] \leq O(\log n) \cdot \mathrm{cost}^s_{\mathcal{A}^*}(R, \mathcal{T}) + O(\log \Delta + \log n) \cdot \mathrm{cost}^t_{\mathcal{A}^*}(R, \mathcal{T}) + \beta(\mathcal{T}) \, ,$$

where $\mathcal{A}^*$ is the projection of $\widetilde{\mathcal{A}}^*$ on $\mathcal{T}$ (that is, same requests are matched at the same time, incurring possibly different space costs). Therefore,

$$
\begin{aligned}
\mathbb{E}_{\mathcal{P},\chi}\left[\text{cost}_{\mathcal{A}}\left(R,\mathcal{M}\right)\right] &\leq \mathbb{E}_{\mathcal{T}\in\mathcal{P}}\left[\mathbb{E}_{\chi}\left[\text{cost}_{\mathcal{A}}\left(R,\mathcal{T}\right)\right]\right] \\
&\leq \mathbb{E}_{\mathcal{T}\in\mathcal{P}}\left[O(\log n)\cdot\text{cost}^s_{\mathcal{A}^*}\left(R,\mathcal{T}\right)+O(\log\Delta+\log n)\cdot\text{cost}^t_{\mathcal{A}^*}\left(R,\mathcal{T}\right)+\beta(\mathcal{T})\right] \\
&= O(\log n)\cdot\mathbb{E}_{\mathcal{T}\in\mathcal{P}}\left[\text{cost}^s_{\mathcal{A}^*}\left(R,\mathcal{T}\right)\right]+O(\log\Delta+\log n)\cdot\text{cost}^t_{\widetilde{\mathcal{A}}^*}\left(R,\mathcal{M}\right)+\beta(\mathcal{M}) \\
&\leq O\left(\log^2 n\right)\cdot\text{cost}^s_{\widetilde{\mathcal{A}}^*}\left(R,\mathcal{M}\right)+O(\log\Delta+\log n)\cdot\text{cost}^t_{\widetilde{\mathcal{A}}^*}\left(R,\mathcal{M}\right)+\beta(\mathcal{M}) \\
&\leq O\left(\log\Delta+\log^2 n\right)\cdot\text{cost}_{\widetilde{\mathcal{A}}^*}\left(R,\mathcal{M}\right)+\beta(\mathcal{M}),
\end{aligned}
$$

where $\beta(\mathcal{M})=\mathbb{E}_{\mathcal{T}\in\mathcal{P}}[\beta(\mathcal{T})]$, the first transition holds since the distance functions in the support of $\mathcal{P}$ dominate δ, the third transition holds since the time costs of $\widetilde{\mathcal{A}}^*$ in $\mathcal{M}$ are the same as those of $\mathcal{A}^*$ in $\mathcal{T}$, and the fourth transition holds by Thm. 2.1.

The remainder of this section is dedicated to the proof of Thm. 4.1 and is organized as follows: First, in Sec. 4.1, we introduce a new stochastic process, called *alternating Poisson process (APP)*, together with some related machinery. APPs play a major role in Sec. 4.2 that forms the heart of the analysis: we prove Thm. 4.1 assuming that online algorithm $\mathcal{A}$ receives a special *end-of-input* signal upon receiving the last request in R and responds to it by immediately matching all remaining active requests. Finally, in Sec. 4.3, we lift the assumption of receiving the end-of-input signal, showing that it does not affect the (multiplicative) competitive ratio.

4.1 Alternating Poisson processes

A major component of the analysis presented in Sec. 4.2 is a stochastic process (more specifically, a point process) that we refer to as an *alternating Poisson process (APP)*. This process is parametrized by its *start time* $t_0\in\mathbb{R}_{\geq 0}$, *length* $\gamma\in\mathbb{R}_{>0}$, *rate* $\lambda\in\mathbb{R}_{>0}$, and a right-continuous *coloring function* $c:[t_0,t_0+\gamma)\to\{1,2,\perp\}$ with finitely many discontinuity points.[7] For simplicity, in the remainder of this section, we assume that the APP starts at time $t_0=0$; this assumption can be lifted by translating any time $t\in[0,\gamma]$ to $t+t_0\in[t_0,t_0+\gamma]$.

Given some $0\leq t\leq t'\leq\gamma$, we define the 1-*volume* and 2-*volume* of the interval $[t,t')$ as

$$
V_1(t,t')=\int_t^{t'}\mathbf{1}(c(x)=1)\,dx
$$

and

$$
V_2(t,t')=\int_t^{t'}\mathbf{1}(c(x)=2)\,dx\,,
$$

respectively. The APP is realized by independent and identically $\text{Exp}(\lambda)$ distributed random variables $Z_1,Z_2,\ldots$ These determine the $[0,\gamma]$-valued random variables $T_1,T_2,\ldots$, referred to as *alter-*

---

[7]The color $\perp$ is redundant for the analysis of the APPs carried out in the present section. We introduce it because it makes things simpler in Sec. 4.2 when we employ the APP framework in the analysis of our online algorithm.

Figure 1: A realization of an alternating Poisson process with time progressing from left to right. The dark gray, light gray, and white intervals represent the colors 1, 2, and $\perp$, respectively. The vertical arrows represent the meaningful alternation times and the horizontal two-sided arrows depict the time intervals that contribute to the digestion of the corresponding iterations.

nation times, defined inductively by fixing $T_0 = 0$ and setting

$$T_j = \begin{cases} \max\left\{t \leq \gamma : V_1(T_{j-1}, t) \leq Z_j\right\}, & j \text{ is odd} \\ \max\left\{t \leq \gamma : V_2(T_{j-1}, t) \leq Z_j\right\}, & j \text{ is even} \end{cases}$$

for $j = 1, 2, \ldots$ Put differently, the alternation times divide the process into *iterations* so that iteration j lasts from time T_{j-1} to time T_j. In odd (resp., even) iterations, the process *digests* the 1s (resp., 2s), ignoring the $\perp$s and the 2s (resp., 1s). If the iteration did not end by time $T_{j-1} < t < \gamma$ and $c(t) = 1$ (resp., $c(t) = 2$), then it ends at time $t + dt$ with probability $\pi = \lambda dt$; the iteration ends at time λ if it did not end beforehand (an illustration is provided in Fig. 1).

The definition of the alternation times implies, in particular, that if $T_{j-1} = \gamma$, then $T_j = \gamma$; we say that the jth alternation time is *meaningful* if $0 < T_j < \gamma$. Observe that if T_j is meaningful and $j \geq 1$ is odd (resp., even), then $c(T_j)$ must be 1 (resp., 2). Let

$$N = \max\{j \in \mathbb{Z}_{\geq 0} \mid T_j < \gamma\}$$

be the random variable counting the number of meaningful alternation times.

Define the $[0, \gamma]$-valued random variables $G_1, G_2, \ldots$ by setting

$$G_j = \begin{cases} V_1(T_{j-1}, T_j), & j \text{ is odd} \\ V_2(T_{j-1}, T_j), & j \text{ is even} \end{cases}$$

and let $G = \sum_{j=1}^{\infty} G_j$. We refer to G_j as the *digestion* of the jth iteration and to G as the *total digestion*.

Lemma 4.2. *For every* $0 \leq t < \gamma$, *we have* $\mathbb{E}[G_j \mid T_{j-1} = t] = \frac{1}{\lambda}\left(1 - e^{-\lambda \cdot V_i(t, \gamma)}\right)$, *where* $i = 1$ *if* j *is odd; and* $i = 2$ *if* j *is even.*[8]

Proof. Assume without loss of generality that j is odd and $i = 1$ (the case that j is even and $i = 2$ is proved following the same line of arguments). The design of the APP implies that conditioned on $T_{j-1} = t$, the random variable G_j satisfies $G_j \sim \min\{\text{Exp}(\lambda), V_1(t, \gamma)\}$, that is, it is distributed

---

[8]Recall that for every $j > 1$, an odd (resp., even) j implies that $c(t) = c(T_{j-1}) = 2$ (resp., $c(t) = c(T_{j-1}) = 1$).

identically to an exponential random variable with rate λ, truncated at $V_1(t, \gamma)$. Fixing $\vartheta = V_1(t, \gamma)$, the assertion follows by observing that

$$
\begin{aligned}
\mathbb{E}\left[\min\left\{\mathrm{Exp}(\lambda), \vartheta\right\}\right] &= \int_0^\vartheta \lambda e^{-\lambda x} x \, dx \; + \; \vartheta e^{-\lambda \vartheta} \\
&= \left. -e^{-\lambda x} x - \frac{1}{\lambda} e^{-\lambda x} \right|_0^\vartheta \; + \; \vartheta e^{-\lambda \vartheta} \\
&= -\vartheta e^{-\lambda \vartheta} - \frac{1}{\lambda} e^{-\lambda \vartheta} + \frac{1}{\lambda} + \vartheta e^{-\lambda \vartheta} \\
&= \frac{1}{\lambda}\left(1 - e^{-\lambda \vartheta}\right),
\end{aligned}
$$

where the second transition is derived using integration by parts with $u(x) = x$ and $v(x) = -e^{-\lambda x}$.
□

Lemma 4.3. $\mathbb{E}[G] = \mathbb{E}[N]/\lambda$.

Proof. Let I_j, $j = 1, 2, \ldots$, be an indicator random variable for the event $T_j < \gamma$ and notice that

$$
\mathbb{E}[N] \; = \; \sum_{j=1}^{\infty} \mathbb{P}\left(N \geq j\right) \; = \; \sum_{j=1}^{\infty} \mathbb{E}\left[I_j\right].
$$

Recalling that

$$
\mathbb{E}[G] \; = \; \sum_{j=1}^{\infty} \mathbb{E}\left[G_j\right],
$$

it suffices to prove that $\mathbb{E}[I_j]/\lambda = \mathbb{E}[G_j]$ for $j = 1, 2, \ldots$ To that end, we show that

$$
\mathbb{E}\left[\mathbb{E}\left[I_j \mid T_{j-1}\right]\right]/\lambda \; = \; \mathbb{E}\left[\mathbb{E}\left[G_j \mid T_{j-1}\right]\right]
$$

which establishes the assertion by the law of total expectation.

The random variable $\mathbb{E}[I_j \mid T_{j-1}]$ maps the event $T_{j-1} = t$ to

$$
\mathbb{E}[I_j \mid T_{j-1} = t] \; = \; \mathbb{P}(\mathrm{Exp}(\lambda) < V_i(t, \gamma)) \; = \; 1 - e^{-\lambda \cdot V_i(t, \gamma)},
$$

where $i = 1$ if j is odd; and $i = 2$ if j is even. The proof is completed by Lem. 4.2 as the random variable $\mathbb{E}[G_j \mid T_{j-1}]$ maps the event $T_{j-1} = t$ to $\mathbb{E}[G_j \mid T_{j-1} = t]$.
□

Lemma 4.4. *The random variable N is stochastically dominated by $1 + 2Z$, where $Z \sim \mathrm{Pois}(\lambda \cdot \min\{V_1(0, \gamma), V_2(0, \gamma)\})$ is a Poisson random variable with parameter $\lambda \cdot \min\{V_1(0, \gamma), V_2(0, \gamma)\}$. Moreover, if K denotes the number of discontinuity points of the coloring function c in $[0, \gamma)$, then $N \leq K + 1$ (with probability 1).*

Proof. Fix $V_1 = V_1(0, \gamma)$ and $V_2 = V_2(0, \gamma)$ and define the random variables

$$
N_1 \; = \; |\{j \in \mathbb{Z}_{\geq 0} \mid T_{2j+1} < \gamma\}| \quad \text{and} \quad N_2 \; = \; |\{j \in \mathbb{Z}_{\geq 1} \mid T_{2j} < \gamma\}|.
$$

The definition of the APP ensures the following four properties:

10

(P1) $N = N_1 + N_2$;

(P2) $N_2 \leq N_1 \leq N_2 + 1$;

(P3) N_i, $i \in \{1, 2\}$, is stochastically dominated by $\mathrm{Pois}(\lambda \cdot V_i)$; and

(P4) N_i, $i \in \{1, 2\}$, is bounded from above by the number of (set-wise) maximal intervals $I \subseteq [0, \gamma)$ satisfying $c(t) = i$ for all $t \in I$.

The second part of the assertion follows directly from properties (P1) and (P4). For the first part, we employ (P1) and (P2) to conclude that $N \leq 1 + 2N_i$ for $i \in \{1, 2\}$. Then, by (P3), it follows that N is stochastically dominated by $1 + 2 \cdot \mathrm{Pois}(\lambda \cdot V_i)$ for $i \in \{1, 2\}$, thus it is stochastically dominated by $1 + 2 \cdot \mathrm{Pois}(\lambda \cdot \min\{V_1, V_2\})$. $\qquad\square$

It will be convenient to also consider a generalization of the APP, referred to as a *rate-varying APP*, in which the fixed rate parameter λ is replaced by a *rate function* $\lambda' : [0, \gamma) \to \mathbb{R}_{>0}$ that may vary in time. This affects the aforementioned iteration termination probability π so that an odd (resp., even) iteration j that did not end by time $T_{j-1} < t < \gamma$, $c(t) = 1$ (resp., $c(t) = 2$), will now end at time $t + dt$ with probability $\pi = \pi(t) = \lambda'(t)dt$. Given some (fixed) $\lambda \in \mathbb{R}_{>0}$, it is straightforward to verify that if the rate function $\lambda'(t)$ is bounded from above by λ, i.e., $\lambda'(t) \leq \lambda$ for all $0 \leq t < \lambda$, then Lem. 4.2 and 4.4 hold also for rate-varying APPs, only that in the former, we should replace the equality in $\mathbb{E}[G_j \mid T_{j-1} = t] = \frac{1}{\lambda}\left(1 - e^{-\lambda \cdot V_i(t, \gamma)}\right)$ with a $\geq$ inequality.

> ***Intuition spotlight:*** *APPs are utilized in the analysis conducted in Sec. 4.2 as they capture the behavior of the stilt-walker algorithm in what can be informally described as "toggling situations". Such situations turn out to appear in multiple parts of the analysis (see Lem. 4.8, 4.10, and 4.12).*

4.2 Analysis under the end-of-input signal assumption

Let $\mathcal{T}$ be an n-point α-HSBT of aspect ratio Δ and let T and $w : T \to \mathbb{R}_{\geq 0}$ be the full binary tree and weight function that realize $\mathcal{T}$. Assume without loss of generality that the minimum positive distance in $\mathcal{T}$ is scaled to 1 so that Δ is the diameter of $\mathcal{T}$.

Our goal in this section is to establish Thm. 4.1 under the end-of-input signal assumption.[9] More formally, assume that the online algorithm is signaled at time $t_{\mathrm{end}} = \max\{t(\rho) \mid \rho \in R\}$ (the arrival time of the last request in R); upon receiving this signal, the algorithm clears the remaining active requests by immediately matching across v for every effective vertex $v \in F(t_{\mathrm{end}})$ (this is guaranteed as the number of active requests at time t_{end} must be even). Let c_{end}^s be the space cost of these matching operations and observe that $c_{\mathrm{end}}^s \leq (n/2) \cdot \Delta$. (Although it does not affect our analysis, it is interesting to point out that c_{end}^s is, in fact, the cost of an optimal matching of the remaining requests.) We start the analysis with the following "warmup" observation regarding the operation of the stilt-walker algorithm.

---

[9]For the convenience of the reader, Fig. 6 provides a schematic overview of the analysis presented in this section.

Observation. *Consider an internal vertex $v \in T - \mathcal{L}$ with children u_1, u_2. The design of $\mathcal{A}$ ensures that:*

1. *the random variable $\mathbf{1}(v \in D(t))$ is independent of the coin tosses of all vertices $u \in T(v)$ (including v);*

2. *$\mathcal{A}$ can match on top of v only when v is odd; and*

3. *if $\mathcal{A}$ matched across or on top of v at time t, then v, u_1, and u_2 are not odd immediately following time t, i.e., $v, u_1, u_2 \notin D(t + dt)$ for infinitesimally small $dt > 0$.*

Proof. To establish property 1, notice that the coin tosses of vertex u determine the decisions of $\mathcal{A}$ to match across u. Matching across u decreases $|C_v(t)|$ by 2, hence it does not affect its parity.

Property 2 is proved by recalling that matching on top of v at time t is realized by matching a request located in some leaf $x \in \mathcal{L}(v)$ to a request located in some leaf $x' \in \mathcal{L} - \mathcal{L}(v)$. Since $v \neq \mathrm{lca}(x, x')$, it must belong to the stilt in $\mathcal{S}(t)$ whose foot is x which establishes the assertion by the definition of $\mathcal{S}(t)$.

Finally, observe that property 3 holds trivially if $\mathcal{A}$ matched across v at time t because this means that $u_1, u_2 \in D(t)$ and thus, $v, u_1, u_2 \notin D(t + dt)$. Otherwise, if $\mathcal{A}$ matched on top of v at time t, then $v \in D(t)$ which means that $u_i \in D(t)$ and $u_{3-i} \notin D(t)$ for some $i \in \{1, 2\}$. This also means that $\mathcal{A}$ matched on top of u_i at time t, therefore $v, u_i, u_{3-i} \notin D(t + dt)$. $\qquad\square$

Intuition spotlight: A key ingredient in the analysis of $\mathcal{A}$'s competitive ratio is an alternative method for measuring its time and space cost on a per-vertex basis. This is facilitated by the definitions of time and space potentials for each internal vertex v.

Time and space potentials. Consider some internal vertex $v \in T - \mathcal{L}$ with children u_1, u_2 and some $0 \leq t_0 < t_1 \leq t_{\mathrm{end}}$. The *time potentials* of v, denoted τ_v and τ_v^*, capture the contributions of v to $\mathrm{cost}_{\mathcal{A}}^t(R, \mathcal{T})$ and $\mathrm{cost}_{\mathcal{A}^*}^t(R, \mathcal{T})$, respectively, in a certain time interval. They are defined by setting

$$\tau_v([t_0, t_1)) = \int_{t_0}^{t_1} \mathbf{1}(v \in F(t)) \, dt \quad \text{and} \quad \tau_v^*([t_0, t_1)) = \int_{t_0}^{t_1} \mathbf{1}(u_1 \in D^*(t)) + \mathbf{1}(u_2 \in D^*(t)) \, dt \, ;$$

in other words, a dt amount is deposited into τ_v whenever $v \in F(t)$ and into τ_v^* whenever $u_i \in D^*(t)$ for $i \in \{1, 2\}$.

The *space potentials* of v, denoted σ_v and σ_v^*, capture the contributions of v to $\mathrm{cost}_{\mathcal{A}}^s(R, \mathcal{T})$ and $\mathrm{cost}_{\mathcal{A}^*}^s(R, \mathcal{T})$, respectively, in a certain time interval. An amount of $w(v)$ is deposited into σ_v whenever $\mathcal{A}$ matches across v; an amount of $w(v)$ is deposited into σ_v^* whenever $\mathcal{A}^*$ matches across or on top of v. In other words, given two requests ρ, ρ' with $x = \ell(\rho)$ and $x' = \ell(\rho')$, if $\mathcal{A}$ matches requests ρ and ρ', then we deposit an amount of $w(u)$ into σ_u for $u = \mathrm{lca}(x, x')$; if $\mathcal{A}^*$ matches requests ρ and ρ', then we deposit an amount of $w(u)$ into σ_u^* for every internal vertex u along the

unique path connecting x and x' in T. Let $\sigma_v([t_0, t_1))$ and $\sigma_v^*([t_0, t_1))$ be the total amount deposited into σ_v and σ_v^*, respectively, during the time interval $[t_0, t_1)$.

For clarity of the exposition, we often write $\tau_v(t_0, t_1)$, $\tau_v^*(t_0, t_1)$, $\sigma_v(t_0, t_1)$, and $\sigma_v^*(t_0, t_1)$ instead of the aforementioned notations. We also extend the definition of these four notations from intervals to collections of disjoint intervals in the natural manner. Thm. 4.1 is established by proving the following three lemmas.

Intuition spotlight: *Lem. 4.5 allows us to express the time and space costs by means of the per-vertex potentials. Lem. 4.6 then means that we can bound the time potential of v under $\mathcal{A}$ by the time and space potentials of v under $\mathcal{A}^*$, charging the extra $w(v)$ on the additive term of the competitive ratio, whereas Lem. 4.7 means that we can bound the space potential of v under $\mathcal{A}$ by its time potential.*

Lemma 4.5. *There exists some $\zeta = \zeta(R)$ such that the time potentials satisfy*

$$\mathrm{cost}_{\mathcal{A}}^t(R, \mathcal{T}) \le \zeta + \sum_{v \in T - \mathcal{L}} O(\tau_v(0, t_{\mathrm{end}})) \quad \text{and} \quad \mathrm{cost}_{\mathcal{A}^*}^t(R, \mathcal{T}) \ge \zeta/h + \sum_{v \in T - \mathcal{L}} \Omega(\tau_v^*(0, t_{\mathrm{end}})/h)$$

(recall that h denotes the height of T). The space potentials satisfy

$$\mathrm{cost}_{\mathcal{A}}^s(R, \mathcal{T}) \le c_{\mathrm{end}}^s + \sum_{v \in T - \mathcal{L}} O(\sigma_v(0, t_{\mathrm{end}})) \quad \text{and} \quad \mathrm{cost}_{\mathcal{A}^*}^s(R, \mathcal{T}) \ge \sum_{v \in T - \mathcal{L}} \Omega((\alpha - 1) \cdot \sigma_v^*(0, t_{\mathrm{end}}))$$

(recall that the parameter α is set in Thm. 4.1).

Lemma 4.6. *For every $v \in T - \mathcal{L}$, it holds that $\mathbb{E}[\tau_v(0, t_{\mathrm{end}})] \le O(\tau_v^*(0, t_{\mathrm{end}}) + \sigma_v^*(0, t_{\mathrm{end}}) + w(v))$.*

Lemma 4.7. *For every $v \in T - \mathcal{L}$, it holds that $\mathbb{E}[\sigma_v(0, t_{\mathrm{end}})] \le \mathbb{E}[\tau_v(0, t_{\mathrm{end}})]$.*

Proof of Lem. 4.5. We first note that

$$\mathrm{cost}_{\mathcal{A}}^t(R, \mathcal{T}) = \sum_{v \in T} \int_0^{t_{\mathrm{end}}} \mathbf{1}(v \in H(t)) \, dt \, .$$

Indeed, as each leaf contains at most one active request, an active request $\rho \in C(t)$ is accounted for in exactly one term of the sum in the RHS of the equation, that is, the term corresponding to the head of the stilt whose foot is $\ell(\rho)$. Since an internal vertex is effective at time t if and only if its two children are in $H(t)$, the last equation can be rewritten as

$$\mathrm{cost}_{\mathcal{A}}^t(R, \mathcal{T}) = \int_0^{t_{\mathrm{end}}} \mathbf{1}(r \in D(t)) \, dt + 2 \cdot \sum_{v \in T - \mathcal{L}} \tau_v(0, t_{\mathrm{end}}) \, .$$

On the other hand, the inequality

$$\mathrm{cost}_{\mathcal{A}^*}^t(R, \mathcal{T}) \ge \frac{1}{h} \cdot \sum_{v \in T} \int_0^{t_{\mathrm{end}}} \mathbf{1}(v \in D^*(t)) \, dt$$

13

holds since each active request under $\mathcal{A}^*$ is accounted for in at most h terms of the sum in the RHS of the inequality, therefore

$$\text{cost}^t_{\mathcal{A}^*}(R, \mathcal{T}) \geq \frac{1}{h} \left(\int_0^{t_{\text{end}}} \mathbf{1}(r \in D^*(t)) \, dt + \sum_{v \in T - \mathcal{L}} \tau^*_v(0, t_{\text{end}}) \right) .$$

The first part of the assertion is established by observing that $r \in D(t)$ if and only if $r \in D^*(t)$, hence we can fix

$$\zeta = \int_0^{t_{\text{end}}} \mathbf{1}(r \in D(t)) \, dt = \int_0^{t_{\text{end}}} \mathbf{1}(r \in D^*(t)) \, dt .$$

The contribution to $\text{cost}^s_{\mathcal{A}}(R, \mathcal{T})$ of matching requests ρ and ρ' by $\mathcal{A}$ is $w(\text{lca}(x, x'))$; this is also its contribution to the space potentials σ, hence

$$\text{cost}^s_{\mathcal{A}}(R, \mathcal{T}) = c^s_{\text{end}} + \sum_{v \in T - \mathcal{L}} \sigma_v(0, t_{\text{end}}) .$$

The contribution to $\text{cost}^s_{\mathcal{A}^*}(R, \mathcal{T})$ of matching requests ρ and ρ' by $\mathcal{A}^*$ is $w(\text{lca}(x, x'))$, whereas since $\mathcal{T} = (T, w)$ is an α-HSBT (recall that $1 < \alpha \leq 2$), its contribution to the space potentials σ^* is bounded from above by $\sum_{i=0}^h w(\text{lca}(x, x')) \cdot (1/\alpha)^i < w(\text{lca}(x, x')) \cdot \alpha/(\alpha - 1)$, hence,

$$\text{cost}^s_{\mathcal{A}^*}(R, \mathcal{T}) \geq \Omega(\alpha - 1) \cdot \sum_{v \in T - \mathcal{L}} \sigma^*_v(0, t_{\text{end}})$$

which completes the proof. $\qquad\square$

Convenient notation. The remainder of this section is dedicated to the proofs of Lem. 4.6 and 4.7. To this end, we fix some internal vertex $v \in T - \mathcal{L}$ with children u_1 and u_2 which facilitates switching to a shorter and simpler notation: Denote $\tau = \tau_v$, $\tau^* = \tau^*_v$, $\sigma = \sigma_v$, and $\sigma^* = \sigma^*_v$. Given some time $t \in [0, t_{\text{end}})$, we write for short

$$X_i(t) = \mathbf{1}(u_i \in D(t)) \qquad X^*_i(t) = \mathbf{1}(u_i \in D^*(t))$$

for $i \in \{1, 2\}$ and

$$X(t) = X_1(t) \oplus X_2(t) \qquad X^*(t) = X^*_1(t) \oplus X^*_2(t) .$$

Notice that $\mathbf{1}(v \in F(t)) = X_1(t) \cdot X_2(t)$ and $\mathbf{1}(u_1 \in D^*(t)) + \mathbf{1}(u_2 \in D^*(t)) = X^*(t) + 2 \cdot X^*_1(t) \cdot X^*_2(t)$, thus

$$\tau(t_0, t_1) = \int_{t_0}^{t_1} X_1(t) \cdot X_2(t) \, dt \quad \text{and} \quad \tau^*(t_0, t_1) = \int_{t_0}^{t_1} X^*(t) + 2 \cdot X^*_1(t) \cdot X^*_2(t) \, dt .$$

It will be convenient to also define

$$Y_i(t) = |\{\rho \in R \mid \ell(\rho) \in \mathcal{L}(u_i) \wedge t(\rho) \leq t\}| \pmod 2$$

for $i \in \{1, 2\}$ and

$$Y(t) = Y_1(t) \oplus Y_2(t) ,$$

observing that the parity of the number of times $\mathcal{A}$ matched on top of u_i (resp., v) up to time t equals $X_i(t) \oplus Y_i(t)$ (resp., $X(t) \oplus Y(t)$).

14

Phases and subphases. We partition the time line $[0, t_{\text{end}})$ into *phases* (defined with respect to v), where each phase is a time interval that starts when the previous phase ends (or at time 0 if this is the first phase) and ends when $\mathcal{A}$ matches on top of v (or at time t_{end} if this is the last phase). A crucial observation is that this partition is fully determined by the coin tosses of $anc(v)$ (namely, the ancestors of v) independently of the coin tosses of v.

We further partition every phase $\phi = [t_0, t_1)$ of v into *subphases*, where each subphase is a time interval that starts when the previous subphase ends (or at time t_0 if this is the first subphase of ϕ) and ends when $\mathcal{A}^*$ matches across *or* on top of v (or at time t_1 if this is the last subphase of ϕ). Notice that matching operations across v performed by $\mathcal{A}$ (fully determined by the coin tosses of v) can occur at the midst of a subphase.

Lemma 4.8. *For every phase $\phi = [t_0, t_1)$ of v, it holds that $\mathbb{E}_v[\sigma(\phi)] = \mathbb{E}_v[\tau(\phi)]$.*

Proof. We investigate the dynamics of $(X_1(t), X_2(t))_{t \in \phi}$ and $(Y_1(t), Y_2(t))_{t \in \phi}$ that take values in $\{0,1\}^2$ (an illustration is provided in Fig. 2). Observe that a new request arriving in $\mathcal{L}(u_i)$, $i \in \{1,2\}$, flips X_i and Y_i without affecting X_{3-i} and Y_{3-i}. While (Y_1, Y_2) is affected only by new request arrivals, the dynamic of (X_1, X_2) is tied to the actions of $\mathcal{A}$ too. Specifically, $\mathcal{A}$ can match across v (recall that $\mathcal{A}$ does not match on top of v in the midst of phase ϕ) only when $(X_1, X_2) = (1,1)$ and if $(X_1, X_2) = (1,1)$ throughout the infinitesimally small time interval $[t - dt, t)$, then $\mathcal{A}$ matches across v at time t with probability $dt/w(v)$ (depending solely on the coin tosses of v), in which case (X_1, X_2) flips to $(X_1(t), X_2(t)) = (0,0)$. Moreover, we know that $(X_1(t_0), X_2(t_0)) = (0,0)$.

Let $(y_1, y_2) = (Y_1(t_0), Y_2(t_0))$. We color the times in ϕ using the coloring function $c : \phi \to \{1, 2, \bot\}$ by setting

$$c(t) = \begin{cases} 1, & (Y_1(t), Y_2(t)) = (\neg y_1, \neg y_2) \\ 2, & (Y_1(t), Y_2(t)) = (y_1, y_2) \\ \bot, & \text{o.w.} \end{cases}$$

The key observation now is that the times at which $\mathcal{A}$ matches across v can be viewed as the meaningful alternation times of an APP Π_ϕ defined over the time interval ϕ with coloring function $c(\cdot)$ and rate $1/w(v)$. (Notice that the role of (y_1, y_2) in the validity of this observation is simply to adjust the dynamic of (X_1, X_2), starting with $(X_1(t_0), X_2(t_0)) = (0,0)$, to the APP framework in which the first digested color is defined to be 1.) Taking N to be the random variable counting the number of meaningful alternation times in Π_ϕ and G to be its total digestion, we conclude that $\sigma(\phi) = w(v) \cdot N$ and $\tau(\phi) = G$. The assertion follows by Lem. 4.3. $\qquad \square$

Fixing the coin tosses in $anc(v)$ and thus, fixing the partition of $[0, t_{\text{end}})$ into phases, we can apply Lem. 4.8 to each individual phase, thus establishing Lem. 4.7 by the linearity of expectation. The remainder of this section is dedicated to proving Lem. 4.6. The first step towards achieving this goal is to bound the time potential of $\mathcal{A}$ per subphase based on the following subphase classification.
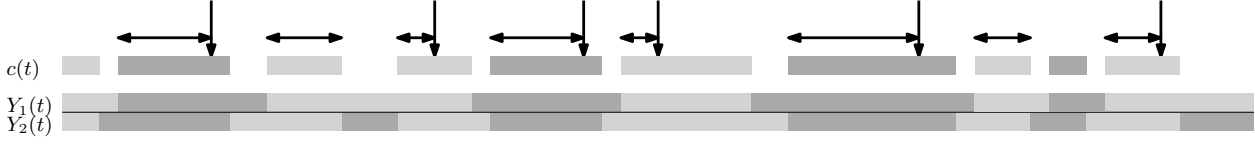
Figure 2: Phase ϕ with time progressing from left to right, assuming that $(y_1, y_2) = (0, 0)$. Bottom rows: the dark gray and light gray intervals represent the times t at which $Y_i(t) = 1$ and $Y_i(t) = 0$, respectively. Top row: the dark gray, light gray, and white intervals represent the times t at which $c(t) = 1$, $c(t) = 2$, and $c(t) = \perp$, respectively. The vertical arrows represent the times at which $\mathcal{A}$ matches across v and the horizontal two-sided arrows depict the time intervals that contribute to $\tau(\phi)$, i.e., when $(X_1, X_2) = (1, 1)$. Notice that towards ϕ's end, we must have $X = X_1 \oplus X_2 = 1$ unless ϕ is the last phase.

0- and 1-subphases. Fix some subphase φ of v. Notice that matching across v (by $\mathcal{A}$) does not affect X_i^*, $i \in \{1, 2\}$, nor does it change $X_1 \oplus X_2$. Thus, there exists some $b = b(\varphi) \in \{0, 1\}$ such that $X_1(t) \oplus X_2(t) \oplus X_1^*(t) \oplus X_2^*(t) = b$ for all $t \in \varphi$; in what follows, we distinguish between two types of subphases: 0-*subphases*, for which $b = 0$, and 1-*subphases*, for which $b = 1$.

Observation 4.9. *If φ is a 1-subphase, then $\tau(\varphi) \leq \tau^*(\varphi)$.*

Proof. Recall that $\tau(\varphi) = \int_\varphi X_1(t) \cdot X_2(t) dt$ and $\tau^*(\varphi) \geq \int_\varphi X_1^*(t) \oplus X_2^*(t) dt$. The assertion follows by the definition of a 1-subphase ensuring that for every $t \in \varphi$, if $(X_1(t), X_2(t)) = (1, 1)$, then $(X_1^*(t), X_2^*(t)) \in \{(0, 1), (1, 0)\}$. $\square$

Lemma 4.10. *If φ is a 0-subphase, then $\mathbb{E}_v[\tau(\varphi)] \leq \tau^*(\varphi) + w(v)$.*

Proof. We investigate the dynamics of $(X_1(t), X_2(t))_{t \in \varphi}$ and $(X_1^*(t), X_2^*(t))_{t \in \varphi}$ that take values in $\{0, 1\}^2$ (an illustration is provided in Fig. 3). By the definition of a 0-subphase, at any time $t \in \varphi$, either $(X_1(t), X_2(t)) = (X_1^*(t), X_2^*(t))$ or $(X_1(t), X_2(t)) = (\neg X_1^*(t), \neg X_2^*(t))$; we refer to the former (resp., latter) as an *agreement* (resp., *disagreement*) state of $\mathcal{A}$ and $\mathcal{A}^*$.

Observe that a new request arriving in $\mathcal{L}(u_i)$, $i \in \{1, 2\}$, flips X_i and X_i^* without affecting X_{3-i} and X_{3-i}^*. While (X_1^*, X_2^*) is affected only by new request arrivals (recall that $\mathcal{A}^*$ does not match across or on top of v in the midst of subphase φ), the dynamic of (X_1, X_2) is tied to the actions of $\mathcal{A}$ too. Specifically, $\mathcal{A}$ can match across v (recall that $\mathcal{A}$ does not match on top of v in the midst of subphase φ) only when $(X_1, X_2) = (1, 1)$ and if $(X_1, X_2) = (1, 1)$ throughout the infinitesimally small time interval $[t - dt, t)$, then $\mathcal{A}$ matches across v at time t with probability $dt/w(v)$ (depending solely on the coin tosses of v), in which case (X_1, X_2) flips to $(X_1(t), X_2(t)) = (0, 0)$, thus toggling the agreement/disagreement state.
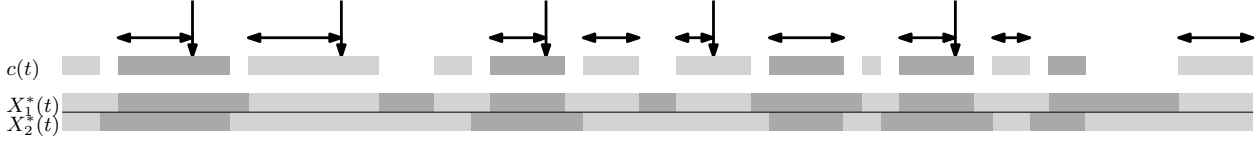
16

Figure 3: Subphase φ with time progressing from left to right, assuming that the subphase starts in an agreement state. Bottom rows: the dark gray and light gray intervals represent the times t at which $X_i^*(t) = 1$ and $X_i^*(t) = 0$, respectively. Top row: the dark gray, light gray, and white intervals represent the times t at which $c(t) = 1$, $c(t) = 2$, and $c(t) = \bot$, respectively. The vertical arrows represent the times at which $\mathcal{A}$ matches across v and the horizontal two-sided arrows depict the time intervals that contribute to $\tau(\phi)$, i.e., when $(X_1, X_2) = (1, 1)$. Notice that by the definition of τ^*, times t at which $X_1^*(t) \oplus X_2^*(t) = 1$ (marked as white intervals in the top row) also contribute to $\tau^*(\phi)$, but this contribution is ignored by our analysis.

Define the functions $c_{\mathrm{agree}} : \varphi \to \{1, 2, \bot\}$ and $c_{\mathrm{disagree}} : \varphi \to \{1, 2, \bot\}$ as follows:

$$
c_{\mathrm{agree}}(t) = \begin{cases} 1, & (X_1^*(t), X_2^*(t)) = (1,1) \\ 2, & (X_1^*(t), X_2^*(t)) = (0,0) \\ \bot, & \text{o.w.} \end{cases} \qquad c_{\mathrm{disagree}}(t) = \begin{cases} 1, & (X_1^*(t), X_2^*(t)) = (0,0) \\ 2, & (X_1^*(t), X_2^*(t)) = (1,1) \\ \bot, & \text{o.w.} \end{cases} .
$$

We color the times in φ using the coloring function $c : \varphi \to \{1, 2, \bot\}$ by setting $c = c_{\mathrm{agree}}$ if the subphase starts in an agreement state; and $c = c_{\mathrm{disagree}}$ if the subphase starts in a disagreement state. The key observation now is that the times at which $\mathcal{A}$ matches across v can be viewed as the meaningful alternation times of an APP Π_φ defined over the time interval φ with coloring function $c(\cdot)$ and rate $1/w(v)$. (Notice that the role of the c_{agree} vs. c_{disagree} distinction in the validity of this observation is simply to adjust the dynamic of (X_1, X_2), starting in an agreement/disagreement state, to the APP framework in which the first digested color is defined to be 1.)

Taking G to be the total digestion of Π_φ, we notice that $\tau(\varphi) = G$. Moreover, the construction of the coloring function $c(\cdot)$ ensures that $\tau^*(\varphi) \geq 2 \int_\varphi X_1^*(t) \cdot X_2^*(t) dt \geq 2 \min\{V_1, V_2\}$, where V_1 and V_2 are the total 1- and 2-volumes of Π_φ, respectively. The assertion follows by Lem. 4.3 and 4.4. $\qquad\qquad\square$

0- and 1-phases. Phase ϕ of v is said to be a 0-*phase* (resp., a 1-*phase*) if it starts with a 0-subphase (resp., a 1-subphase). Let P^0 (resp., P^1) be the set of 0-phases (resp., 1-phases) of v. Using Obs. 4.9 and Lem. 4.10, we establish Lem. 4.6 (our goal in the remainder of this section) by proving the following inequalities:

$$\mathbb{E}_{v, anc(v)} \left[\tau(P^0) \right] \leq O\left(\tau^*(0, t_{\mathrm{end}}) + \sigma^*(0, t_{\mathrm{end}}) + w(v) \right) \tag{1}$$

$$\mathbb{E}_{v, anc(v)} \left[\tau(P^1) \right] \leq O\left(\tau^*(0, t_{\mathrm{end}}) + \sigma^*(0, t_{\mathrm{end}}) \right) . \tag{2}$$

17

Lem. 4.11 (a combination of Obs. 4.9 and Lem. 4.10 essentially) plays an important role in the desired proofs.

Lemma 4.11. *If ϕ is a 0-phase, then*

$$\mathbb{E}_v\left[\tau(\phi)\right] \leq \tau^*(\phi) + 2\sigma^*(\phi) + w(v);$$

if ϕ is a 1-phase, then

$$\mathbb{E}_v\left[\tau(\phi)\right] \leq \tau^*(\phi) + 2\sigma^*(\phi).$$

Proof. Let $U^b(\phi)$ be the set of b-subphases of ϕ for $b \in \{0,1\}$. If $U^0(\phi) = \emptyset$ and $U^1(\phi) = \{\varphi\}$, then we can employ Obs. 4.9 to conclude that $\mathbb{E}_v[\tau(\phi)] \leq \tau^*(\varphi) = \tau^*(\phi)$. If $U^0(\phi) = \{\varphi\}$ and $U^1(\phi) = \emptyset$, then we can employ Lem. 4.10 to conclude that $\mathbb{E}_v[\tau(\phi)] \leq w(v) + \tau^*(\varphi) = w(v) + \tau^*(\phi)$.

Since all but the last subphases of ϕ end when $\mathcal{A}^*$ matches across or on top of v, it follows by the definition of σ^* that $|U^0(\phi) \cup U^1(\phi)| = 1 + \sigma^*(\phi)/w(v)$. Therefore, if $|U^0(\phi) \cup U^1(\phi)| > 1$, then we can employ Obs. 4.9 and Lem. 4.10 to conclude that

$$\begin{aligned}
\mathbb{E}_v\left[\tau(\phi)\right] &\leq \sum_{\varphi \in U^0(\phi)} (w(v) + \tau^*(\varphi)) + \sum_{\varphi \in U^1(\phi)} \tau^*(\varphi) \\
&= \tau^*(\phi) + |U^0(\phi)| \cdot w(v) \\
&\leq \tau^*(\phi) + |U^0(\phi) \cup U^1(\phi)| \cdot w(v) \\
&= \tau^*(\phi) + (1 + \sigma^*(\phi)/w(v)) \cdot w(v) \\
&= \tau^*(\phi) + w(v) + \sigma^*(\phi) \leq \tau^*(\phi) + 2 \cdot \sigma^*(\phi),
\end{aligned}$$

where the last transition holds since $|U^0(\phi) \cup U^1(\phi)| > 1$ implies that $\sigma^*(\phi) \geq w(v)$. The assertion follows. $\square$

Fixing the coin tosses in $anc(v)$ (and thus, fixing the partition of $[0, t_{\text{end}})$ into phases), we can apply Lem. 4.11 to each individual 1-phase, hence obtaining (2) by the linearity of expectation.

> **Intuition spotlight:** *It remains to establish (1) which turns out to be more demanding: for 0-phases ϕ, the upper bound on $\mathbb{E}_v[\tau(\phi)]$ promised by Lem. 4.11 includes an additive $w(v)$ term and we have to make sure that it does not dominate the $\tau^*(\phi)$ and $\sigma^*(\phi)$ terms too often. This is done via a classification of the phases with respect to their starting time.*

Early and late phases. Recall the definition of $Y(t) = Y_1(t) \oplus Y_2(t)$ and let t_{late} be the smallest $t \in [0, t_{\text{end}})$ such that $\min\{\int_t^{t_{\text{end}}} Y(s)ds, \int_t^{t_{\text{end}}} \neg Y(s)ds\} \leq w(v)$. Phase ϕ with starting time t is said to be an *early* phase if $t < t_{\text{late}}$ and a *late* phase if $t \geq t_{\text{late}}$. (Intuitively, this means that when an early phase starts, we still have more than $w(v)$ time units of $Y(t) = 0$ and more than $w(v)$ time units of $Y(t) = 1$.) Let P_{early} and P_{late} be the sets of early and late phases, respectively. Let K be the number of discontinuity points of $Y(t)$ in the interval $[0, t_{\text{late}})$.

We would like to take a closer look at the partition of $[0, t_{\text{end}})$ into phases. To that end, consider some phase ϕ with starting time T^- and end time T^+. Fixing $T^- = t$ for some $t \in [0, t_{\text{end}})$, the end time T^+ is a random variable fully determined by the coin tosses in $anc(v)$ after time t. An important property of this random variable is cast in the following lemma (together with two other important properties of the partition of $[0, t_{\text{end}})$ into phases).

Lemma 4.12. *The partition of $[0, t_{\text{end}})$ into phases satisfies the following three properties:*
(P1) if $t < t_{\text{late}}$, then $\mathbb{E}_{anc(v)} \left[\int_{T^-}^{T^+} X(s)ds \mid T^- = t \right] \geq w(v)(1 - 1/e)$;
(P2) $|P_{\text{early}}| \leq K + 1$; and
(P3) $\mathbb{E}_{anc(v)}[|P_{\text{late}}|] = O(1)$ with an exponentially vanishing upper tail.

Proof. We investigate the dynamics of $(X(t))_{t \in [0, t_{\text{end}})}$ and $(Y(t))_{t \in [0, t_{\text{end}})}$ (an illustration is provided in Fig. 4). A new request arriving in $\mathcal{L}(v)$ flips X and Y. While Y is affected only by new request arrivals, the dynamic of X is tied to the actions of $\mathcal{A}$ too. Specifically, the design of the stilt-walker algorithm ensures that $\mathcal{A}$ can match on top of v only when $X = 1$ (recall that matching across v does not affect the partition of $[0, t_{\text{end}})$ to phases). Suppose that $X = 1$ throughout the infinitesimally small time interval $I = [t - dt, t)$; let S be the stilt in $\mathcal{S}(t')$ to which v belongs for all $t' \in I$ and let $v' \in anc(v)$ be the head of S. Then $\mathcal{A}$ matches across v' and on top of v at time t with probability $\pi(t) = dt/w(v')$ (depending solely on the coin tosses of v'), in which case X flips to $X(t) = 0$. Since v' is an ancestor of v, we know that $\pi(t) < dt/w(v)$.

We color the time line using the coloring function $c : [0, t_{\text{end}}) \rightarrow \{1, 2, \perp\}$ by setting

$$c(t) = \begin{cases} 1, & Y(t) = 1 \\ 2, & Y(t) = 0 \end{cases}$$

(note that $\perp$, whose preimage under c is empty, is included in the range of c for the sake of compatibility with the APP framework). The key observation now is that the times at which $\mathcal{A}$ matches on top of v can be viewed as the meaningful alternation times of a rate-varying APP $\Pi_{[0, t_{\text{end}})}$ defined over the time interval $[0, t_{\text{end}})$ with coloring function $c(\cdot)$ and rate function bounded from above by $1/w(v)$ (recall that a rate-varying APP is a generalization of an APP defined in the end of Sec. 4.1 of the full version).

Taking G_ϕ to be the digestion of the iteration in $\Pi_{[0, t_{\text{end}})}$ that starts at time $T^- = t$, we notice that $\int_{T^-}^{T^+} X(s)ds = G_\phi$; recalling that the definition of t_{late} guarantees that $\min \left\{ \int_t^{t_{\text{end}}} \mathbf{1}(c(t) = 1)dt, \int_t^{t_{\text{end}}} \mathbf{1}(c(t) = 2)dt \right\} > w(v)$ for every $t < t_{\text{late}}$, we obtain property (P1) by applying (the rate-varying version of) Lem. 4.2 to $\Pi_{[0, t_{\text{end}})}$. Property (P2) holds simply by applying Lem. 4.4 to the $[0, t_{\text{late}})$-restriction of $\Pi_{[0, t_{\text{end}})}$. To obtain property (P3), we consider the $[t_{\text{late}}, t_{\text{end}})$-restriction of $\Pi_{[0, t_{\text{end}})}$, denote its number of meaningful alternation times by N, and observe that $|P_{\text{late}}|$ is stochastically dominated by $N + 1$; the property then follows by Lem. 4.4 since $\min \left\{ \int_{t_{\text{late}}}^{t_{\text{end}}} \mathbf{1}(c(t) = 1)dt, \int_{t_{\text{late}}}^{t_{\text{end}}} \mathbf{1}(c(t) = 2)dt \right\} \leq w(v)$. $\qquad \square$

Figure 4: Interval $[0, t_{\text{end}})$ with time progressing from left to right. The dark gray and light gray intervals represent the times t at which $Y(t) = 1$ and $Y(t) = 0$, respectively. The solid vertical arrows represent the times at which $\mathcal{A}$ matches on top of v. The dashed vertical arrow represent time t_{late}.

Corollary 4.13. *If ϕ is a 0-phase that starts at time $T^- = t < t_{\text{late}}$, then $\mathbb{E}_{anc(v)}[\tau^*(\phi) + \sigma^*(\phi) \mid T^- = t] \geq \Omega(w(v))$.*

Proof. As T^+ is a random variable fully determined by the coin tosses in $anc(v)$ after time t, $\tau^*(\phi)$ and $\sigma^*(\phi)$ are also random variables fully determined by the coin tosses in $anc(v)$ after time t. Since $\tau^*(\phi) \geq \int_t^{T^+} X^*(s)ds$ and since ϕ starts with a 0-subphase φ during which $X = 1$ implies $X^* = 1$, the assertion follows from Lem. 4.12(P1), recalling that if ϕ contains any subphase other than φ (in particular, a 1-subphase during which $X = 1$ does not imply $X^* = 1$), then $\sigma^*(\phi) \geq w(v)$. $\square$

We are now ready to establish (1). This is done by defining $P^0_{\text{early}} = P^0 \cap P_{\text{early}}$ and $P^0_{\text{late}} = P^0 \cap P_{\text{late}}$ to be the sets of early and late 0-phases, respectively, and proving the following two lemmas.

Lemma 4.14. $\mathbb{E}_{v,anc(v)}\left[\tau(P^0_{\text{early}})\right] = O\left(\tau^*(0, t_{\text{end}}) + \sigma^*(0, t_{\text{end}})\right)$.

Proof. Lem. 4.12(P2) ensures that $|P^0_{\text{early}}| \leq K + 1$. Let $\phi_1, \ldots, \phi_{K+1}$ be the sequence of early 0-phases, where, for the sake of the analysis, we introduce a suffix of empty *dummy* phases so that each dummy phase ϕ_j, $|P^0_{\text{early}}| + 1 \leq j \leq K + 1$, starts and ends at some arbitrary dummy time $\hat{t} > t_{\text{end}}$, thus ensuring that $\tau(\phi_j) = \tau^*(\phi_j) = \sigma^*(\phi_j) = 0$.

Fix some $1 \leq j \leq K + 1$ and let T^- and T^+ be the random variables that capture the starting time and end time of ϕ_j. We argue that

$$\mathbb{E}_{v,anc(v)}\left[\tau(\phi_j)|T^- = t\right] \leq O\left(\mathbb{E}_{anc(v)}\left[\tau^*(\phi_j) + \sigma^*(\phi_j) \mid T^- = t\right]\right) \tag{3}$$

for any t in the support of T^-. This clearly holds if ϕ_j is an empty dummy phase (which means that $t = \hat{t}$), so assume that $t < t_{\text{late}}$. Consider the random variable $\mathbb{E}_v[\tau(\phi_j)|T^- = t, T^+]$ that maps the event $T^+ = s$ (defined over the coin tosses in $anc(v)$) to $\mathbb{E}_v[\tau(\phi_j)|T^- = t, T^+ = s]$. By Lem. 4.11, the latter satisfies $\mathbb{E}_v[\tau(\phi_j)|T^- = t, T^+ = s] \leq \tau^*(t, s) + 2\sigma^*(t, s) + w(v)$. Therefore,

$$\mathbb{E}_{anc(v)}\left[\mathbb{E}_v\left[\tau(\phi_j)|T^- = t, T^+\right]\right] \leq \mathbb{E}_{anc(v)}\left[\tau^*(\phi_j) + 2\sigma^*(\phi_j) \mid T^- = t\right] + w(v)$$
$$\leq O\left(\mathbb{E}_{anc(v)}\left[\tau^*(\phi_j) + \sigma^*(\phi_j) \mid T^- = t\right]\right),$$

where the last transition follows from Cor. 4.13, thus establishing (3) by the law of total expectation.

20

Consider the random variable $\mathbb{E}_{v,anc(v)}[\tau(\phi_j)|T^-]$ that maps the event $T^- = t$ (defined over the coin tosses in $anc(v)$) to $\mathbb{E}_{v,anc(v)}[\tau(\phi_j)|T^- = t]$. Using the bound provided for the latter by (3) and applying the law of total expectation, we conclude that

$$\mathbb{E}_{v,anc(v)}\left[\tau(\phi_j)\right] = \mathbb{E}_{anc(v)}\left[\mathbb{E}_{v,anc(v)}\left[\tau(\phi_j)|T^-\right]\right] \leq O\left(\mathbb{E}_{anc(v)}\left[\tau^*(\phi_j) + \sigma^*(\phi_j)\right]\right).$$

Therefore, by the linearity of expectation, we derive

$$\mathbb{E}_{v,anc(v)}\left[\tau(P_{\text{early}}^0)\right] = \sum_{j=1}^{K+1} \mathbb{E}_{v,anc(v)}\left[\tau(\phi_j)\right]$$

$$\leq \sum_{j=1}^{K+1} O\left(\mathbb{E}_{anc(v)}\left[\tau^*(\phi_j) + \sigma^*(\phi_j)\right]\right) = O\left(\mathbb{E}_{anc(v)}\left[\tau^*(P_{\text{early}}^0) + \sigma^*(P_{\text{early}}^0)\right]\right)$$

which establishes the assertion. $\square$

Lemma 4.15. $\mathbb{E}_{v,anc(v)}\left[\tau(P_{\text{late}}^0)\right] \leq O\left(\tau^*(0, t_{\text{end}}) + \sigma^*(0, t_{\text{end}}) + w(v)\right)$.

Proof. Conditioned on $|P_{\text{late}}^0| = m$, Lem. 4.11 guarantees that

$$\mathbb{E}_v\left[\tau(P_{\text{late}}^0)\right] \leq \tau^*(P_{\text{late}}^0) + 2\sigma^*(P_{\text{late}}^0) + m \cdot w(v) \leq \tau^*(P_{\text{late}}) + 2\sigma^*(P_{\text{late}}) + m \cdot w(v).$$

By Lem. 4.12(P3),

$$\mathbb{E}_{anc(v)}[|P_{\text{late}}^0|] \leq \mathbb{E}_{anc(v)}[|P_{\text{late}}|] \leq O(1)$$

with an exponentially vanishing upper tail, thus

$$\mathbb{E}_{v,anc(v)}\left[\tau(P_{\text{late}}^0)\right] \leq O\left(\tau^*(P_{\text{late}}) + \sigma^*(P_{\text{late}}) + w(v)\right)$$

which establishes the assertion. $\square$

4.3 Lifting the end-of-input signal assumption

We now turn to lift the end-of-input signal assumption, showing that Thm. 4.1 holds also without it. Recall that $t_{\text{end}} = \max\{t(\rho) \mid \rho \in R\}$ denotes the arrival time of the last request in R and let $C = C(t_{\text{end}})$ and $F = F(t_{\text{end}})$ be the set of remaining active requests and the set of effective vertices at time t_{end}, respectively. The analysis presented in Sec. 4.2 relies on the assumption that upon receiving the end-of-input signal at time t_{end}, the algorithm immediately clears all the requests in C by matching across every vertex in F which contributes $c_{\text{end}}^s = \sum_{v \in F} w(v)$ to the space cost of $\mathcal{A}$ (this contribution to the space cost of $\mathcal{A}$ is taken into account in Sec. 4.2).

An examination of the matching policy of the stilt-walker algorithm reveals that in reality, the requests in C are indeed cleared by matching across the vertices in F, only that these matching operations are not performed immediately at time t_{end}, but rather at slightly later (random) times,

thus introducing an additional contribution to the time cost of $\mathcal{A}$. Specifically, taking $\rho, \rho' \in C$ to be the supporting requests of some effective vertex $v \in F$, notice that on expectation, $\mathcal{A}$ matches across v at time $t_{\text{end}} + w(v)$ which accounts for an additional contribution of a $2w(v)$ term to the algorithm's expected time cost. Summing over all vertices in F, we conclude that by adding

$$2 \sum_{v \in F} w(v) \; < \; 2 \sum_{v \in T} w(v)$$

to the β term in Thm. 4.1, we can lift the end-of-input assumption as promised.

5 A fixed penalty for clearing requests

In this section, we consider the online *MPMDfp* problem: a variant of MPMD in which the algorithm is allowed to clear any request $\rho \in R$ at time $t \geq t(\rho)$ without matching it to another request, incurring a fixed penalty $p > 0$ (a parameter of the problem), on top of the time cost $t - t(\rho)$ of ρ, that adds to its total cost. Notice that in contrast to MPMD, the MPMDfp problem is well defined also for odd values of $|R|$.

Theorem 5.1. *There exists a randomized online MPMDfp algorithm for $\mathcal{M}$ whose competitive ratio is $O\left(\log^2 n + \log \Delta\right)$, where n is the number of points in the underlying metric space and Δ is its aspect ratio.*

Proof. Consider the underlying n-point metric space $\mathcal{M} = (V, \delta)$ and let $d = \min_{x \neq y \in V} \delta(x, y)$ and $D = \max_{x \neq y \in V} \delta(x, y)$ be the minimum and maximum distances between any two distinct points in $\mathcal{M}$, respectively, so that the aspect ratio of $\mathcal{M}$ is $\Delta = D/d$. Assume for the time being that the penalty p satisfies $d/2 < p < 2D$.

Let $\widehat{\mathcal{M}} = (V \times \{1, 2\}, \widehat{\delta})$ be the metric space defined by setting

$$\widehat{\delta}((x, i_x), (y, i_y)) = \delta(x, y) + p \cdot |i_x - i_y|$$

for every $x, y \in V$ and $i_x, i_y \in \{1, 2\}$. The assumption that $d/2 < p < 2D$ implies that the aspect ratio of $\widehat{\mathcal{M}}$ is proportional to Δ. Let $\widehat{R} = \{\rho_1, \rho_2 \mid \rho \in R\}$, where ρ_i, $i \in \{1, 2\}$, is defined by setting $t(\rho_i) = t(\rho)$ and $\ell(\rho_i) = (\ell(\rho), i)$.

We construct an online MPMDfp algorithm $\mathcal{A}_{\text{fp}}$ with the desired competitive ratio from the stilt-walker algorithm $\mathcal{A}$ as follows. Algorithm $\mathcal{A}_{\text{fp}}$ simulates $\mathcal{A}$ on $\widehat{\mathcal{M}}, \widehat{R}$ and handles the requests in R according to the actions of $\mathcal{A}$ on the requests in $\widehat{R}$. Specifically, for every $\rho \in R$, if $\mathcal{A}$ matches ρ_1 to some request ρ_1', located in $V \times \{1\}$, at time t, then $\mathcal{A}_{\text{fp}}$ matches ρ to ρ' at time t; if $\mathcal{A}$ matches ρ_1 to some request ρ_2', located in $V \times \{2\}$, at time t, then $\mathcal{A}_{\text{fp}}$ clears ρ without matching it (paying the fixed p-penalty) at time t.

The design of $\mathcal{A}_{\text{fp}}$ and the fact that $\widehat{\delta}((x, 1), (y, 2)) \geq p$ for every $x, y \in V$ guarantee that

$$\text{cost}_{\mathcal{A}_{\text{fp}}}(R, \mathcal{M}) \leq \text{cost}_{\mathcal{A}}(\widehat{R}, \widehat{\mathcal{M}}). \tag{4}$$

22

Moreover, the construction of $\widehat{\mathcal{M}}$ and $\widehat{R}$ ensures that if $\mathcal{A}^*$ is an optimal offline MPMD algorithm and $\mathcal{A}^*_{\mathrm{fp}}$ is an optimal offline MPMDfp algorithm, then

$$\mathrm{cost}_{\mathcal{A}^*}(\widehat{R}, \widehat{\mathcal{M}}) \leq 2 \cdot \mathrm{cost}_{\mathcal{A}^*_{\mathrm{fp}}}(R, \mathcal{M}) \tag{5}$$

since $\mathcal{A}^*$ can project the actions of $\mathcal{A}^*_{\mathrm{fp}}$ on each side of $\widehat{\mathcal{M}}$, matching ρ_1 to ρ_2 whenever $\mathcal{A}^*_{\mathrm{fp}}$ clears ρ without matching it. The assertion follows since the stilt-walker algorithm $\mathcal{A}$ is $O(\log^2 n + \log \Delta)$-competitive for the MPMD problem.

Now, if $p < d/2$, then an MPMDfp (online or offline) algorithm is always better off clearing the requests by paying the fixed penalty than by matching them. Therefore, in this case, the MPMDfp problem over $\mathcal{M}$ can be decomposed into n independent instances of the MPMDfp over a 1-point metric space. Each such instance (essentially a repeated version of the ski rental problem) admits an $O(1)$-competitive online algorithm, thus so does the whole problem.

It remains to consider the case where $p > 2D$. In this case, we construct the metric space $\widehat{\mathcal{M}}$ slightly differently: first employ Thm. 2.1 to probabilistically embed $\mathcal{M}$ in a $(1+\Omega(1/\log n))$-HSBT (T, w); then, take two copies of T, call them T_1 and T_2, and connect them so that their roots become the children of a new root $\widehat{r}$, extending the weight function w by setting $w(\widehat{r}) = p$. Notice that the resulting metric space is also a $(1 + \Omega(1/\log n))$-HSBT whose point set can be renamed $V \times \{1, 2\}$ so that (x, i) is a leaf of T_i for every $x \in V$ and $i \in \{1, 2\}$. The rest of the construction of $\mathcal{A}_{\mathrm{fp}}$ is unchanged.

Although the aspect ratio of the metric space $\widehat{\mathcal{M}}$ in this case may be large (as large as p/d), notice that the height of the underlying $(1 + \Omega(1/\log n))$-HSBT is still $O(\log \Delta + \log n)$, where Δ is the aspect ratio of $\mathcal{M}$. This establishes the assertion by recalling that the $\log \Delta$ term in the competitive ratio of the stilt-walker algorithm comes from an upper bound on the height of its HSBT. $\qquad \square$

6   The deterministic version of the stilt-walker algorithm

In this section, we consider the deterministic version of the stilt-walker algorithm, denoted $\mathcal{A}_d$, obtained by replacing the $(1/w(v))$-rate exponential timer maintained at each internal vertex $v \in T - \mathcal{L}$ with a deterministic $w(v)$-timer. In other words, the matching policy of $\mathcal{A}_d$ is similar to that of $\mathcal{A}$ with one difference: If the last time $\mathcal{A}_d$ matched across v was at time t_0 (take $t_0 = 0$ if $\mathcal{A}_d$ still has not matched across v), then the next time it matches across v is the minimum t_1 that satisfies

$$\int_{t_0}^{t_1} \mathbf{1}(v \in F(t)) \, dt = w(v) \, .$$

Theorem 6.1. *The competitive ratio of $\mathcal{A}_d$ on n-point $(1 + \Omega(1/\log n))$-HSBTs is $\Omega(n)$.*

Proof. Let n be some large power of 2 and let T be an n-leaf perfect binary tree (with all leaves at depth $\lg n$). Let $w : T \to \mathbb{R}_{\geq 0}$ be the weight function defined by setting $w(x) = 0$ for every leaf x; and $w(v) = (1 + 1/\lg n)^{\lg(n)-1-i}$ for every internal vertex v of depth i. Consider the HSBT $\mathcal{T} = (T, w)$ and notice that the distance between any two distinct points in $\mathcal{T}$ is $\Theta(1)$. We name some of the internal vertices and subtrees of T according to the labels in Fig. 5.

Let $\mathcal{A}_d^*$ denote the benchmark offline algorithm and take ϵ to be a small positive real. For every subtree T_j, $j = 1, \ldots, 6$, fix some arbitrary leaf x_j and consider the following scenario Γ (refer to Fig. 5 for an illustration):

- 2 requests arrive at time 0 at leaves x_1 and x_6 (one each). $\mathcal{A}_d^*$ immediately matches these requests. Following that, the sole effective vertex of $\mathcal{A}_d$ is v_1 with supporting leaves x_1 and x_6.
- 4 requests arrive at time $w(v_1) - \epsilon$ at leaves x_2, x_3, x_4, and x_5 (one each). Following that, the effective vertices of $\mathcal{A}_d$ are:
  v_1 with supporting leaves x_3 and x_4;
  v_3 with supporting leaves x_1 and x_2; and
  v_5 with supporting leaves x_5 and x_6.
- The timer of v_1 expires at time $w(v_1)$ and $\mathcal{A}_d$ matches (across v_1) the active requests hosted at leaves x_3 and x_4.
- 4 requests arrive at time $w(v_1) + \epsilon$ at leaves x_2, x_3, x_4, and x_5 (one each). Both $\mathcal{A}_d$ and $\mathcal{A}_d^*$ immediately match the request pairs hosted at x_2 and x_5; $\mathcal{A}_d^*$ also immediately matches the request pairs hosted at x_3 and x_4. Following that, the effective vertices of $\mathcal{A}_d$ are:
  v_2 with supporting leaves x_1 and x_3; and
  v_4 with supporting leaves x_4 and x_6.

Consider the subscenario Γ' induced on Γ by the time interval $(0, w(v_1)+\epsilon]$. The key observation is that at the beginning of Γ', $\mathcal{A}_d$ had 2 active requests located at leaves whose LCA is the the root of T (v_1), whereas at its end, $\mathcal{A}_d$ has 4 active requests located at leaves whose LCAs are the two depth 1 vertices (v_2 and v_4). On the other hand, $\mathcal{A}_d^*$ started and ended subscenario Γ' with no active requests, paying a total cost of $O(\epsilon)$ during that time period.

Subscenarios analogous to Γ' are now applied in a recursive manner to the subtrees rooted at the depth i vertices of T, for $i = 1, \ldots, \lg(n) - 3$. This results in $\mathcal{A}_d$ having active requests at exactly $n/2$ distinct leaves of T; to clear all of them, $\mathcal{A}_d$ will have to pay $\Omega(n)$ in space cost. On the other hand, the total cost payed by $\mathcal{A}_d^*$ during all these applications is $O(\epsilon n)$ which can be made arbitrarily small. Adding the $O(1)$ space cost payed by $\mathcal{A}_d^*$ at time 0 for matching the first two requests (across v_1), we conclude that the competitive ratio of $\mathcal{A}_d$ is $\Omega(n)$, as promised. $\qquad\square$
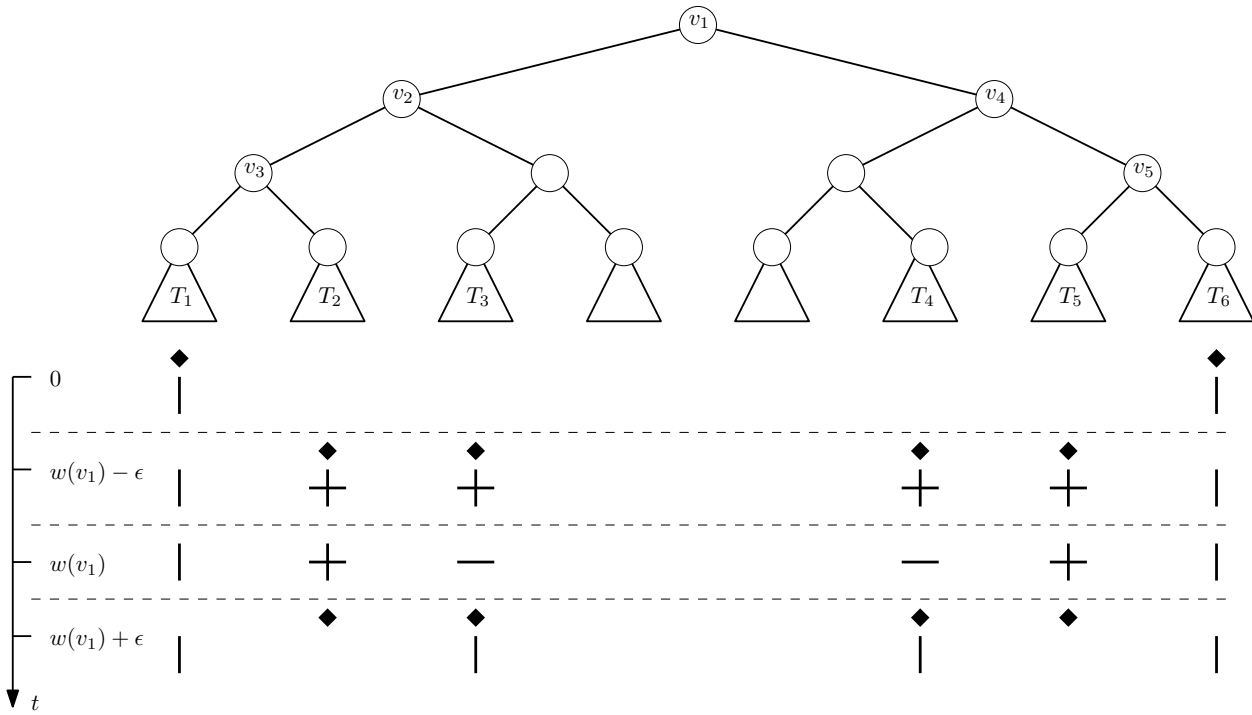
Figure 5: The perfect binary tree T and scenario Γ at times of interest featured on the left. For every $j = 1, \ldots, 6$ and for every time t, a diamond shape depicts a request arriving at leaf x_j at time t; a vertical segment depicts an active request under $\mathcal{A}_d$ at leaf x_j at time t; and a horizontal segment depicts an active request under $\mathcal{A}_d^*$ at leaf x_j at time t.

References

[1] G. Aggarwal, G. Goel, C. Karande, and A. Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1253–1264, 2011.

[2] I. Averbakh and M. Baysan. Approximation algorithm for the on-line multi-customer two-level supply chain scheduling problem. *Operations Research Letters*, 41(6):710 – 714, 2013.

[3] I. Averbakh and Z. Xue. On-line supply chain scheduling problems with preemption. *European Journal of Operational Research*, 181(1):500 – 504, 2007.

[4] B. Awerbuch, Y. Azar, S. Leonardi, and O. Regev. Minimizing the flow time without migration. *SIAM J. Comput.*, 31(5):1370–1382, 2002.

[5] B. Awerbuch, S. Kutten, and D. Peleg. Competitive distributed job scheduling (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 571–580, 1992.

[6] Y. Azar, A. Epstein, L. Jez, and A. Vardi. Make-to-order integrated scheduling and distribution. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 140–154, 2016.

[7] N. Bansal, N. Buchbinder, A. Gupta, and J. Naor. A randomized o(log2 k)-competitive algorithm for metric bipartite matching. *Algorithmica*, 68(2):390–403, 2014.

[8] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193, 1996.

[9] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 161–168, 1998.

[10] Y. Bartal. Graph decomposition lemmas and their role in metric embedding methods. In *Proceedings of ESA, 12th Annual European Symposium on Algorithms*, pages 89–97, 2004.

[11] B. E. Birnbaum and C. Mathieu. On-line bipartite matching made simple. *SIGACT News*, 39(1):80–87, 2008.

[12] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, New York, NY, USA, 1998.

[13] N. Buchbinder, K. Jain, and J. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of ESA, 15th Annual European Symposium on Algorithms*, pages 253–264, 2007.

[14] A. Cote, A. Meyerson, and L. J. Poplawski. Randomized k-server on hierarchical binary trees. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 227–234, 2008.

[15] N. R. Devanur, K. Jain, and R. D. Kleinberg. Randomized primal-dual analysis of RANK-ING for online bipartite matching. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 101–107, 2013.

[16] D. R. Dooly, S. A. Goldman, and S. D. Scott. TCP dynamic acknowledgment delay: Theory and practice (extended abstract). In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 389–398, 1998.

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

[18] J. Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69:125–130, 1965.

[19] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[20] Y. Emek, T. Langner, and R. Wattenhofer. The price of matching with metric preferences. http://ie.technion.ac.il/~yemek/Publications/pmmp.pdf. Extended abstract appeard in Proceedings of ESA 2015.

[21] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004.

[22] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 982–991, 2008.

[23] B. Kalyanasundaram and K. Pruhs. Online weighted matching. *J. Algorithms*, 14(3):478–488, 1993.

[24] B. Kalyanasundaram and K. Pruhs. The online transportation problem. *SIAM J. Discrete Math.*, 13(3):370–383, 2000.

[25] A. R. Karlin, C. Kenyon, and D. Randall. Dynamic TCP acknowledgement and other stories about e/(e-1). In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing*, pages 502–509, 2001.

[26] A. R. Karlin, M. S. Manasse, L. A. McGeoch, and S. Owicki. Competitive randomized algorithms for non-uniform problems. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 301–309, 1990.

[27] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator. Competitive snoopy caching. In *27th Annual Symposium on Foundations of Computer Science*, pages 244–254, 1986.

[28] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 352–358, 1990.

[29] S. Khuller, S. G. Mitchell, and V. V. Vazirani. On-line algorithms for weighted bipartite matching and stable marriages. *Theor. Comput. Sci.*, 127(2):255–267, 1994.

[30] A. Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[31] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. Adwords and generalized on-line matching. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 264–273, 2005.

[32] A. Meyerson, A. Nanavati, and L. J. Poplawski. Randomized online algorithms for minimum metric bipartite matching. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*, *SODA*, 2006.

[33] S. Miyazaki. On the advice complexity of online bipartite matching and online stable marriage. *Inf. Process. Lett.*, 114(12):714–717, 2014.

[34] J. Naor and D. Wajc. Near-optimum online ad allocation for targeted advertising. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC*, pages 131–148, 2015.

[35] B. Patt-Shamir. Private communication, 2015.

[36] E. M. Reingold and R. E. Tarjan. On a greedy heuristic for complete matching. *SIAM J. Comput.*, 10(4):676–681, 1981.

A   Probabilistic embedding of arbitrary metric spaces in HSBTs

Our goal in this section is to prove Thm. 2.1. The main ingredient in this proof is the following celebrated theorem of Fakcharoenphol et al. [21].

Theorem A.1 ([21])**.** *Consider some n-point metric space (V, δ) and let $\mathcal{U}$ be the set of all 2-HSTs over V with distance functions $\delta_{\mathcal{T}}$ that dominate δ in the sense that $\delta_{\mathcal{T}}(x, y) \geq \delta(x, y)$ for every $x, y \in V$. There exists a probability distribution $\mathcal{P}$ over $\mathcal{U}$ such that $\mathbb{E}_{(V, \delta_{\mathcal{T}}) \in \mathcal{P}}[\delta_{\mathcal{T}}(x, y)] \leq O(\log n) \cdot \delta(x, y)$ for every $x, y \in V$. Moreover, the probability distribution $\mathcal{P}$ can be sampled efficiently.*

Observe that by the definition of HSTs, the rooted trees realizing the 2-HSTs promised by Thm. A.1 are of height $O(\log \Delta)$, where $\Delta = \frac{\max_{x,y \in V} \delta(x,y)}{\min_{x,y \in V} \delta(x,y)}$ is the aspect ratio of the metric space (V, δ). These rooted trees have arbitrary degrees, whereas Thm. 2.1 requires rooted trees with degrees at most 2. We resolve this obstacle with the help of the following lemma, proved by Patt-Shamir [35].

Lemma A.2 ([35])**.** *Consider some n-leaf rooted tree T. There exist a (rooted) full binary tree T' and an injection $f : T \to T'$ such that*

(1) $f(v)$ is an ancestor of $f(u)$ in T' if and only if v is an ancestor of u in T;

(2) $\mathrm{depth}_{T'}(f(v)) \leq \mathrm{depth}_{T'}(f(p^T(v))) + O(\log n)$, where $\mathrm{depth}_{T'}(\cdot)$ denotes the depth operator in tree T'; and

(3) $\mathrm{height}(T') = O(\mathrm{height}(T) + \log n)$.

Consider some n-point tree metric space (T, w) in the support of the probability distribution promised by Thm. A.1 and let T' and $f : T \to T'$ be the full binary tree and injection obtained by applying Lem. A.2 to T. We construct a weight function $w' : T' \to \mathbb{R}_{\geq 0}$ on the vertices of T' by first setting $w'(v) = 2 \cdot w(f^{-1}(v))$ for every $v \in f(T)$, and then fixing $w'(v) = w'(p^{T'}(v))/(1 + \Omega(1/\log n))$ for every $v \notin f(T)$. Lem. A.2 guarantees that (T', w') is a $(1 + \Omega(1/\log n))$-HSBT. Taking δ and δ' to be the distance functions of (T, w) and (T', w'), respectively, we observe that

$$\delta(x, y) \leq \delta'(x, y) \leq 2\delta(x, y)$$

for every two points x, y in the metric space(s), thus establishing Thm. 2.1.
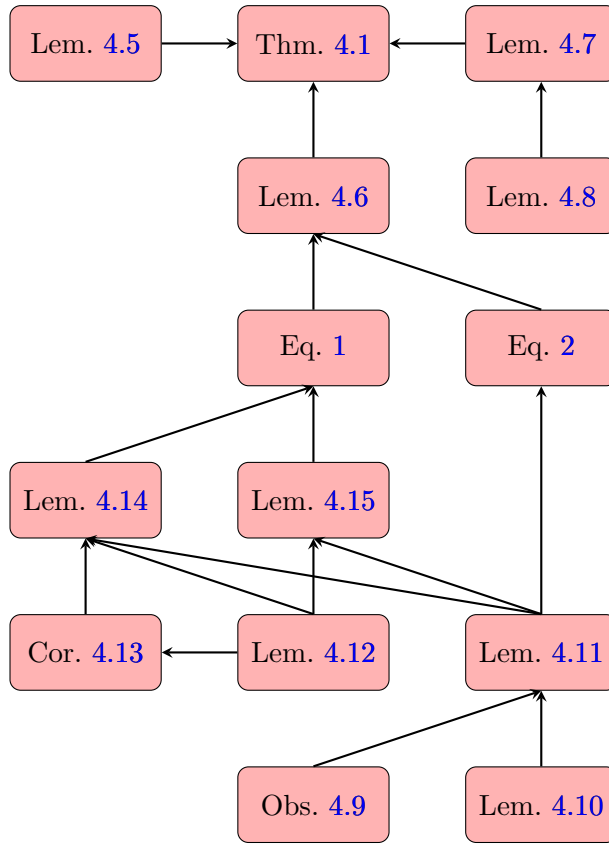
Figure 6: A schematic overview of the analysis carried out in Sec. 4.2, depicting the interdependencies between its components. An arrow pointing from A to B indicates that the proof corresponding to B depends on the statement corresponding to A. The proofs of Lem. 4.8, 4.10, and 4.12 are based on the APP machinery developed in Sec. 4.1.

Notation	Definition	Defined on page
$\ell(\rho)$	location of ρ	1
$t(\rho)$	arrival time of ρ	1
$\mathrm{cost}^s(\rho)$	space cost of ρ	1
$\mathrm{cost}^t(\rho)$	time cost of ρ	1
$\mathrm{cost}^s(R)$	space cost of R	2
$\mathrm{cost}^t(R)$	time cost of R	2
$\mathrm{cost}(R, \mathcal{M})$	total cost	2
$p(v)$	parent of v	4
$T(v)$	subtree rooted at v	4
$\mathcal{L}(v)$	leaves of $T(v)$	4
$anc(v)$	ancestors of v	4
$\mathrm{depth}(v)$	depth of v	4
$\mathrm{height}(T)$	height of T	4
$\mathrm{lca}(x, y)$	least common ancestor of x and y	4
$C_v(t)$	set of requests with locations in $\mathcal{L}(v)$ active at time t	5
$C(t)$	$C_v(t)$ for $v = r$	5
$D(t)$	set of odd vertices at time t	5
$D^*(t)$	set of vertices odd under $\mathcal{A}^*$ at time t	5
$\mathcal{S}(t)$	set of stilts induced by the odd vertices at time t	5
$H(t)$	set of heads of the stilts in $\mathcal{S}(t)$	5
$F(t)$	set of effective vertices at time t	5
t_{end}	arrival time of the last request	11
c^s_{end}	space cost of matching the active requests at time t_{end}	11
$\tau_v(t_0, t_1)$	time potential v accumulates during $[t_0, t_1)$ under $\mathcal{A}$	12
$\tau^*_v(t_0, t_1)$	time potential v accumulates during $[t_0, t_1)$ under $\mathcal{A}^*$	12
$\sigma_v(t_0, t_1)$	space potential v accumulates during $[t_0, t_1)$ under $\mathcal{A}$	12
$\sigma^*_v(t_0, t_1)$	space potential v accumulates during $[t_0, t_1)$ under $\mathcal{A}^*$	12
$X_i(t)$	$\mathbf{1}(u_i \in D(t))$	14
$X^*_i(t)$	$\mathbf{1}(u_i \in D^*(t))$	14
$X(t)$	$X_1(t) \oplus X_2(t)$	14
$X^*(t)$	$X^*_1(t) \oplus X^*_2(t)$	14
$Y_i(t)$	$\|\{\rho \in R \mid \ell(\rho) \in \mathcal{L}(u_i) \wedge t(\rho) \le t\}\| \pmod 2$	14
$Y(t)$	$Y_1(t) \oplus Y_2(t)$	14
P^0	set of 0-phases	17
P^1	set of 1-phases	17
t_{late}	smallest t s.t. $\min\left\{\int_t^{t_{\mathrm{end}}} Y(t)dt, \int_t^{t_{\mathrm{end}}} \neg Y(t)dt\right\} \le w(v)$	18
P_{early}	set of early phases	18
P_{late}	set of late phases	18
K	number of discontinuity points of $Y(t)$ in $[0, t_{\mathrm{late}})$	18

Table 1: A table of notations.