

Online and Incremental Fractional Vertex Cover on Trees

Júlia Baligács 

Department of Computer Science,
University of Oxford, UK

Bartłomiej Bosek 

Institute of Theoretical Computer Science,
Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland

Yann Disser 

Department of Mathematics,
TU Darmstadt, Germany

Andreas Emil Feldmann 

Department of Computer Science,
University of Sheffield, UK

Grzegorz Gutowski 

Institute of Theoretical Computer Science,
Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland

Katarzyna Kępińska 

Institute of Theoretical Computer Science,
Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland

Paweł Putra 

Institute of Informatics,
University of Warsaw, Poland

Anna Zych-Pawlewicz 

Institute of Informatics,
University of Warsaw, Poland

Abstract

In this paper we study the fractional vertex cover problem on trees in two related models: online and incremental. In the online model, the vertices of the tree are known a priori and the edges arrive one at a time. The goal is to maintain a fractional vertex cover of the tree, i.e., an assignment of fractional weights from $[0, 1]$ to the vertices such that the weights of endpoints of every edge sum up to at least one. After each edge arrival, we need to modify the fractional vertex cover to cover the new edge as well. However, we can only increase the values assigned to vertices. The problem was studied before (in the vertex arrival model) by Wang and Wong, who motivated it as a generalization of the ski-rental problem, but also (more importantly) by its close connection to the dual online matching problem. They presented a 1.901-competitive algorithm for general graphs in the vertex arrival model. We present an $\frac{11}{6} \approx 1.83$ -competitive algorithm for trees in the more general edge arrival model.

In addition, we study the fractional vertex cover problem in an incremental model, where we again seek a fractional vertex cover after every update, but all the updates to the tree are known to the algorithm a priori. In this model, we give a 1.5-competitive algorithm and provide a matching lower bound.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics; Theory of computation → Online algorithms

Keywords and phrases fractional vertex cover, online algorithms, incremental algorithms, edge arrival model, trees, competitive analysis

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.158

Funding This work is a result of research conducted within research project number 2023/51/B/ST6/02833 financed by the National Science Centre, Poland (Bartłomiej Bosek, Grzegorz Gutowski, Katarzyna Kępińska, Paweł Putra, and Anna Zych-Pawlewicz)

Júlia Baligács: During employment in Warsaw supported by ERC grant BOBR (grant no. 948057). During employment in Oxford supported by ERC grant CCOO (grant no. 101165139).



© Júlia Baligács, Bartłomiej Bosek, Yann Disser, Andreas Emil Feldmann, Grzegorz Gutowski, Katarzyna Kępińska, Paweł Putra, and Anna Zych-Pawlewicz;
licensed under Creative Commons License CC-BY 4.0

34th Annual European Symposium on Algorithms (ESA 2026).

Editors: Philip Bille, Seth Pettie, and Sabine Storandt; Article No. 158; pp. 158:1–158:15



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction and Related Work

In this paper we consider the fractional vertex cover problem in the online and in the incremental edge arrival model. In this model, edges of an underlying graph appear one by one and we need to maintain a fractional vertex cover after each step. Here, a fractional (resp. integral) vertex cover is an assignment of weights from $[0, 1]$ (resp. from $\{0, 1\}$) to the vertices so that every edge in the graph is *covered*, i.e., such that the weights assigned to its endpoints sum to at least 1. The objective is to minimize the total weight of the vertices, and the challenge is that our decisions are irrevocable in the sense that we may only ever increase the weights assigned to the vertices from one step to the next. We cannot hope to maintain an optimal fractional vertex cover in each time step, instead we aim for solutions of small *competitive ratio*, i.e., solutions that are never suboptimal by more than this ratio.

The online setting differs from the incremental setting in that it assumes no prior knowledge of the edges of the graph, while the incremental setting assumes that we know the graph a priori. Both settings have in common that we need to define increasing sequences of vertex weights that simultaneously approximate a minimum fractional vertex cover across all time steps. The incremental setting can be viewed as an online problem, where the only unknown is the number of steps that we need to optimize for (much like in the ski rental problem). We provide online algorithms for both the online and the incremental setting in the edge arrival model for the case where the underlying graph is a tree.

The online fractional vertex cover problem was studied on general graphs by Wang and Wong [30], but in the vertex arrival model, where vertices are revealed one at a time together with the adjacent edges to the previously revealed vertices. This model is easier than the edge arrival model in the sense that vertex addition can be simulated by adding adjacent edges one by one. Wang and Wong motivated the problem as a generalization of the ski-rental problem, and, more importantly, by its connection to the fractional online matching problem. They present a 1.901-competitive online algorithm maintaining a fractional vertex cover, with a proof based on primal-dual analysis. This means that the authors maintain an online solution to the dual fractional matching problem. By weak duality, the maintained matching is $(1/1.901 \approx 0.526)$ -competitive. This lower bound was later matched by an algorithm by Tang and Zhang [29], which was later shown to be optimal by Tang [28].

The existence of a $(1/2 + \varepsilon)$ -competitive algorithm for the fractional matching problem on general graphs in vertex arrival model had been a long standing open question. Thus, as a byproduct of their approach, Wang and Wong provide such an online algorithm. Building on this result, Gamlath et al. [13] used a general rounding technique to transform an online fractional matching algorithm to an online randomized (integral) matching algorithm. They obtain a randomized $(\frac{1}{2} + \varepsilon)$ -competitive online matching algorithm for some small $\varepsilon > 0$. On the other hand, Gamlath et al. also show that, in the edge arrival model, no online algorithm (even randomized) can beat a competitive ratio of $1/2$. This is an important result, since online matching is a very well-motivated and widely studied problem. A straightforward greedy algorithm is $1/2$ -competitive for this problem, and a significant amount of research has been devoted to providing better ratios in different regimes and for different graph classes [19, 22, 26, 6, 3, 20, 4].

For forests, the online fractional matching problem was studied by Buchbinder et al. [4]. They show an online $5/9$ -competitive fractional matching algorithm which translates to a randomized integral algorithm with the same expected competitive ratio. Their analysis relies again on the primal-dual technique, where instead of maintaining a proper fractional vertex cover throughout the runtime of the algorithm, a new one is constructed retrospectively after

each edge introduction. The construction of those covers heavily depends on imposing a parent-child relation edge endpoints imposed by rooting each tree in the forest arbitrarily at each timestep. Sadly this does not yield a $(9/5 = 1.8)$ -competitive online fractional vertex cover, as merging two trees in the forest by introducing an edge between them and rooting this tree can effectively reverse the relationship between some edge endpoints. This in turn leads to a vertex weight decreasing in the cover at some steps, which is not allowed in an online setting. They also show an upper bound of ≈ 0.5914 on the competitive ratio of any fractional matching algorithms in this model. Online fractional matchings on forests were also studied by Jiang and Zhang [17] in the free disposal model, in which one each of the matched edges can be later removed forever by the algorithm. They show a $(5/8 = 0.625)$ -competitive algorithm for online fractional matching in this setting, showing a clear separation between competitive ratios of algorithms attainable for the two models. Importantly the free disposal model allows them to maintain a single dual solution throughout the algorithm by taking advantage of the model allowing non-incremental changes, which is not possible in our case.

To the best of our knowledge, the best known competitive ratio for fractional vertex cover on trees in the edge arrival model is the trivial ratio of 2, as the bound of Wang and Wong [30] does not apply to the edge arrival model. In this paper, we improve the state of the art by presenting a $(\frac{11}{6} \approx 1.83)$ -competitive online fractional vertex cover algorithm for trees.

We also present a 1.5-competitive incremental fractional vertex cover algorithm for trees, as well as a matching lower bound for the problem. The fractional vertex cover problem does not seem to have been previously considered from an incremental perspective. Incremental minimization problems studied in the past include k -median [5, 25, 21], facility location [27, 21], and k -center [15]. In all of these problems, solutions are only constrained in their (growing) cardinality, otherwise every solution is feasible and the structure of the problem is encoded in the objective. In contrast, in the incremental vertex cover problem, we have a covering constraint in the sense that only solutions covering all the edges are permitted. As noted by Lin et al. [21], the results for the minimum latency problem in [2, 14] implicitly yield results for the incremental k -MST problem, which is another example of an incremental minimization problem with a non-trivial feasibility constraint, namely that the solution must form a tree.

There is a substantial body of work on incremental *maximization* problems, ranging from problems under a growing cardinality constraint [1, 10, 11] to growing knapsack [24, 9, 8] and matroid [12, 18] constraints. Hassin and Rubinfeld [16] considered the incremental maximum matching problem, which is related to our setting via the duality between matchings and vertex covers. They show that the best possible competitive ratio for incremental matchings is $\sqrt{2}$, which was later complemented by Matuschke, Skutella, and Soto [23] with a tight bound of $\ln(4) \approx 1.38$ for the randomized competitive ratio. The setting was further generalized to matroid intersection [12] and independence systems with bounded exchangeability [18]. That being said, these results do not seem transferable to (fractional) vertex cover.

2 Online Fractional Vertex Cover Algorithm

In this section, we present our online fractional vertex cover algorithm for trees that we call RGB algorithm. Before we proceed, we more formally introduce the necessary definitions.

A *fractional vertex cover* of a graph $G = (V, E)$ is a function $f : V \rightarrow [0, 1]$ such that, for every edge $(u, v) \in E$, we have $f(u) + f(v) \geq 1$. The online edge arrival model assumes that the edges of G are revealed one at a time according to some *presentation order*

$e_1 \ll e_2 \ll \dots \ll e_m$. We denote by $G_t := (V, \{e_1, \dots, e_t\})$ the graph after presenting all edges e_i with $i \leq t$. The task of an online algorithm is to produce a sequence $f_1, \dots, f_m$ of fractional vertex covers such that:

- (1) For all $t \in \{1, \dots, m\}$, f_t is a fractional vertex cover for G_t .
- (2) For all $t \in \{2, \dots, m\}$ and $v \in V$, we have $f_{t-1}(v) \leq f_t(v)$, i.e., the covers are increasing.

Importantly, the algorithm needs to output f_t without any knowledge of $G_{t+1}, \dots, G_m$. By OPT_t , we denote the weight of a minimum vertex cover in G_t . For a fixed algorithm ALG , by $\text{ALG}_t := \sum_{v \in V} f_t(v)$, we denote the total weight of the fractional vertex cover generated by it at time t . We say that the algorithm ALG is ρ -competitive if $\max\{\frac{\text{ALG}_t}{\text{OPT}_t} \mid t \in [m]\} \leq \rho$ for any input. As the algorithm has no knowledge of the future, we can equivalently say that an algorithm is ρ -competitive if $\text{ALG}_m \leq \rho \cdot \text{OPT}_m$ for every instance on m edges. When t is clear from context we omit the subscript and also use y_v as a shorthand of $f_t(v)$. If for an edge $(u, v) \in E$ it holds that $y_u + y_v \geq 1$, we say that (u, v) is *covered*. If, on the other hand, $y_u + y_v < 1$, we say that (u, v) is *uncovered* with a *deficit* of $d = 1 - y_u - y_v$.

A *fractional matching* for a graph G is an assignment of weights $m_{uv} \in [0, 1]$ to each edge $(u, v) \in E$ such that for every $v \in V$ it holds that $\sum_{(u,v) \in E} m_{uv} \leq 1$, i.e., the sum of weights on edges incident to any vertex does not exceed 1. In the fractional matching problem, the objective is to maximize the sum of the values m_{uv} assigned to the edges.

It is worth noting that on bipartite graphs (and in particular on trees) both the fractional vertex cover problem and the fractional matching problem admit optimal integral solutions. In particular, OPT_t will always be integral for our purposes. We also note that it is very easy to obtain a competitive ratio of 2 for online fractional vertex cover problem: we maintain a maximal matching and pick the endpoints of the matching edges to the vertex cover (i.e., assign them the values 1). Clearly, the size of a minimum vertex cover is at least the size of the matching, and hence 2-competitiveness follows.

We now move on to defining the RGB algorithm.

► **Definition 1** (RGB algorithm). *The RGB algorithm colors each arriving edge of the graph by one of three available colors from the list [**red** = 1, **green** = 2, **blue** = 3]. On arrival of an edge e , it is assigned color c_e , which is either the first color from the list not present on its incident edges, or **blue** if no such color exists (see Algorithm 1). Let $C(v)$ be a set of colors present on the edges adjacent to v . The weight of a vertex in the cover is assigned according to the colors of adjacent edges in the following way*

$$f(v) = \begin{cases} 1 & \text{if } C(v) = \{\text{red}, \text{green}, \text{blue}\} \\ \frac{5}{6} & \text{if } C(v) = \{\text{red}, \text{green}\} \\ \frac{4}{6} & \text{if } C(v) = \{\text{red}, \text{blue}\} \\ \frac{2}{6} & \text{if } C(v) = \{\text{green}, \text{blue}\} \\ \frac{3}{6} & \text{if } C(v) = \{\text{red}\} \\ \frac{1}{6} & \text{if } C(v) = \{\text{green}\} \\ 0 & \text{if } C(v) = \{\text{blue}\} \text{ or } C(v) = \emptyset. \end{cases}$$

We start the analysis by proving the correctness of RGB algorithm.

► **Lemma 2.** *The RGB Algorithm maintains a fractional vertex cover after every edge arrival.*

Proof. We need to prove that after introducing an online edge $e = (u, v)$ it holds that $y_u + y_v \geq 1$. We make case distinction for each possible color assigned to the new edge e :

■ **Algorithm 1** RGB Algorithm.

Input: Online graph $G = (V, E)$ with offline vertices V
Output: A fractional vertex cover of G

- 1 Initialize $y_u \leftarrow 0$ for each $u \in V$
- 2 **for** each online edge $e = (u, v)$ **do**
- 3 $c_e \leftarrow \min \{ \{i \in \{1, 2\} \mid u, v \text{ are not adjacent to any edge of color } i\} \cup \{3\} \}$
- 4 $y_u \leftarrow f(u)$
- 5 $y_v \leftarrow f(v)$
- 6 **end**
- 7 **return** $\{x_v\}$ for all $v \in V$

- $c_e = \text{red}$: Both u and v are incident to a **red** edge, implying that $y_u, y_v \geq \frac{1}{2}$, so $y_u + y_v \geq 1$.
- $c_e = \text{green}$: Since $e = (u, v)$ got assigned **green**, it means that at least one of u, v must be adjacent to a **red** edge. Assume without loss of generality that it is u , then $\{\text{red}, \text{green}\} \subseteq C(u)$ giving $y_u \geq \frac{5}{6}$, and $\text{green} \in C(v)$ giving $y_v \geq \frac{1}{6}$, so $y_u + y_v \geq 1$
- $c_e = \text{blue}$: This means that at least one of u, v is adjacent to a **red** edge and at least one is adjacent to a **green** edge. There are two cases:
 - Case 1: At least one of u, v is adjacent to both a **red** and a **green** edge. Assume without loss of generality that u is such a vertex. Then $\{\text{red}, \text{green}, \text{blue}\} \subseteq C(u)$ giving $y_u \geq 1$, so $y_u + y_v \geq 1$.
 - Case 2: Both u and v are adjacent to exactly one of red and green. Assume without loss of generality that u is adjacent to **red** and v is adjacent to **green**. Then we have $\{\text{red}, \text{blue}\} \subseteq C(u)$, giving $y_u \geq \frac{4}{6}$ and $\{\text{green}, \text{blue}\} \subseteq C(v)$ giving $y_v \geq \frac{2}{6}$, so $y_u + y_v \geq 1$ ◀

Next, we prove that the RGB algorithm achieves the desired competitive ratio.

► **Theorem 3.** *The RGB Algorithm is $\frac{11}{6}$ -competitive on forests.*

Proof. For a fixed moment t we will show how to transform the total weight of the cover ALG_t produced up to this moment into a fractional matching M of weight $\frac{6}{11} \text{ALG}_t$. The result will follow from weak duality, as the vertex cover LP is the dual of the matching LP (see for instance [7]).

Let m_{uv} denote the weight discharged onto the edge (u, v) . To create the matching we first root each connected component at an arbitrary vertex. Then for each vertex v we discharge the weight y_v by doing the following.

1. If vertex v has a red edge (u_r, v) , then increase $m_{u_r v}$ by Δ_{red} and decrease y_v by Δ_{red} , where $\Delta_{\text{red}} = \min(y_v, \frac{4}{6} - \frac{1}{6}[u_r \text{ is a parent of } v])$.
2. If vertex v has a green edge (u_g, v) , then increase $m_{u_g v}$ by Δ_{green} and decrease y_v by Δ_{green} , where $\Delta_{\text{green}} = \min(y_v, \frac{2}{6} - \frac{1}{6}[u_g \text{ is a parent of } v])$.
3. If vertex v has a blue edge (u_b, v) , then increase $m_{u_b v}$ by Δ_{blue} and decrease y_v by Δ_{blue} , where $\Delta_{\text{blue}} = \min(y_v, \frac{1}{6} - \frac{1}{6}[u_b \text{ is a parent of } v])$.

To prove that this is enough to discharge the entirety of y_v , it suffices to show that the sum of discharged amounts is at least y_v for all possible sets $C(v)$ of colors incident to v . For $C(v) = \{\text{red}, \text{green}, \text{blue}\}$, by summing the discharged amounts over all steps, we obtain that the discharged amount is at least

$$\min \left(y_v, \frac{4}{6} + \frac{2}{6} + \frac{1}{6} - \frac{1}{6} [\text{some } u \in \{u_r, u_g, u_b\} \text{ is a parent of } v] \right) = \min \left(y_v, \frac{6}{6} \right) = y_v,$$

as $y_v \leq 1$ and each vertex v has at most one parent. For $C(v) = \{\text{red}, \text{green}\}$ we get

$$\min \left(y_v, \frac{4}{6} + \frac{2}{6} - \frac{1}{6} [\text{some } u \in \{u_r, u_g\} \text{ is a parent of } v] \right) = \min \left(y_v, \frac{5}{6} \right) = y_v,$$

as $y_v \leq \frac{5}{6}$. For $C(v) = \{\text{red}, \text{blue}\}$ we get

$$\min \left(y_v, \frac{4}{6} + \frac{1}{6} - \frac{1}{6} [\text{some } u \in \{u_r, u_b\} \text{ is a parent of } v] \right) = \min \left(y_v, \frac{4}{6} \right) = y_v,$$

as $y_v \leq \frac{4}{6}$. For $C(v) = \{\text{green}, \text{blue}\}$ we get

$$\min \left(y_v, \frac{2}{6} + \frac{1}{6} - \frac{1}{6} [\text{some } u \in \{u_g, u_b\} \text{ is a parent of } v] \right) = \min \left(y_v, \frac{2}{6} \right) = y_v,$$

as $y_v \leq \frac{2}{6}$. For $C(v) = \{\text{red}\}$ we get

$$\min \left(y_v, \frac{4}{6} - \frac{1}{6} [u_r \text{ is a parent of } v] \right) = \min \left(y_v, \frac{3}{6} \right) = y_v,$$

as $y_v \leq \frac{3}{6}$. For $C(v) = \{\text{green}\}$ we get

$$\min \left(y_v, \frac{2}{6} - \frac{1}{6} [u_g \text{ is a parent of } v] \right) = \min \left(y_v, \frac{1}{6} \right) = y_v,$$

as $y_v \leq \frac{1}{6}$. For $C(v) = \{\text{blue}\}$ we have $y_v = 0$ so there is nothing to discharge. Let m_v denote the sum of weights of edges incident to vertex v . Clearly v can be adjacent to at most one **red** and at most one **green** edge. Additionally, a vertex may be adjacent to multiple **blue** edges, but only one of them can have a positive weight. To prove this, consider two cases:

- (1) v has a **blue** edge to its parent p : In this case both **red** and **green** edges of v point to its children, so the entirety of y_v will be discharged onto them. This means that the only **blue** edge with positive charge can be the edge (v, p) , as **blue** edges can only get charge from their parent vertex.
- (2) v has no **blue** edge to its parent: Since **blue** edges can only get charge from their parent vertex, and v charges only one **blue** edge, there is only one edge with positive charge incident to v .

Summing the contributions of edges around v we get

$$m_v \leq \underbrace{\frac{4}{6} + \frac{3}{6}}_{\text{red}} + \underbrace{\frac{2}{6} + \frac{1}{6}}_{\text{green}} + \max \left(\underbrace{\frac{1}{6} + 0}_{\text{blue (1)}}, \underbrace{0 + \frac{1}{6}}_{\text{blue (2)}} \right) = \frac{7}{6} + \frac{3}{6} + \frac{1}{6} = \frac{11}{6},$$

where under-braced parts show the contributions of each edge as a sum of contribution of an edge of given color from its parent and child end in the tree. This gives us that $M = \left\{ \frac{6}{11} m_{uv} | (u, v) \in E \right\}$ is a fractional matching.

Let $|M^*|$ be the weight of a maximum fractional matching. From weak duality we get

$$\frac{6}{11} \text{ALG} = |M| \leq |M^*| = \text{OPT}$$

$$\text{ALG} \leq \frac{11}{6} \text{OPT}$$

◀

3 Incremental Vertex Cover

In this section, we study a variant of the online vertex cover problem in which the algorithm has full knowledge of the future, and the objective is to maintain a good solution at all times. The motivation for this model is the following. Consider the process of building a road network. During construction, the site must be monitored by security cameras. Given the full construction plan, we must decide where and when to install cameras. Once installed, they cannot be moved, and at every point in time we aim to minimize the total number of cameras in use.

We formalize this setting as the *incremental vertex cover* problem. Similar to the online setting, we are given as input a graph $G = (V, E)$ together with an ordering of its edges $e_1 \ll \dots \ll e_m$. The goal is to compute an increasing sequence $f_1, \dots, f_m$ of fractional vertex covers such that:

- (1) For all $t \in \{1, \dots, m\}$, f_t is a fractional vertex cover for $G_t = (V, \{e_1, \dots, e_t\})$.
- (2) For all $t \in \{2, \dots, m\}$ and $v \in V$, we have $f_{t-1}(v) \leq f_t(v)$, i.e., the covers are increasing.

We denote $f_t(v)$ as $y_v^{(t)}$. The key difference to the online setting is that the algorithm has full knowledge of G and the ordering of the edges $e_1 \ll \dots \ll e_m$. The challenge here is to compute a good solution for all times. As in the online setting, we are seeking algorithms with a small competitive ratio, which is defined by $\max_{t \in [m]} \frac{\text{ALG}_t}{\text{OPT}_t}$.

In this section, we focus on the fractional incremental vertex cover problem on forests. We show that the optimal competitive ratio is exactly $3/2$. Specifically, we present an algorithm achieving this ratio and a matching lower bound. The algorithm is half-integral and follows a simple local rule. We conjecture that the same ratio of $3/2$ is achievable in the integral setting. However, we construct an example showing that no purely local rule can achieve a competitive ratio of $3/2$, illustrating that the integral case is substantially more challenging.

Our lower bound construction is based on a simple instance – a binary tree revealed layer by layer. Despite its simplicity, the analysis is technically involved, as it must capture the behavior of an arbitrary fractional algorithm. The key idea is to derive constraints on the average weight assigned to each layer over time and to use LP duality to show that no algorithm satisfying these constraints can achieve a competitive ratio better than $3/2$.

3.1 A 1.5-competitive half-integral algorithm

In this subsection, we introduce a half-integral algorithm for incremental vertex cover, called **TWOCHILDRULE**, and prove that it achieves a competitive ratio of $3/2$.

Before describing the algorithm, let us first establish a useful property. Observe that larger instances are more difficult to solve in the following sense: For a forest F , let $\rho(F)$ denote the best achievable competitive ratio over all instances defined on F . If F is a subgraph of F' , then $\rho(F) \leq \rho(F')$: Indeed, any instance on F can be embedded into F' by revealing all edges in $E(F)$ first and postponing all edges in $E(F') \setminus E(F)$ to the end. As a consequence, we can assume wlog. that F is a tree (instead of only a forest), for which we can pick a root, and we can assume that every vertex has either at least two children or is a leaf.

The algorithm **TWOCHILDRULE** is formally described in Algorithm 2. Throughout the description, we use the convention that whenever $y_v^{(t)}$ is not explicitly specified, it remains unchanged, i.e., $y_v^{(t)} = y_v^{(t-1)}$. The algorithm exploits the parent–child structure of the tree and follows a simple local rule. Vertices with two children are eventually assigned weight 1.

When an edge $e = (p, c)$ arrives, where c is the first child of p , the algorithm looks ahead to the next time step at which either p or c receives an additional child. If p receives another child before c does, then p is assigned weight 1. Otherwise, both p and c are assigned weight $1/2$.

■ **Algorithm 2** TWOCHILDRE.

```

1 root the final tree arbitrarily;
2 initialize  $y_v^{(0)} = 0$  ( $v \in V$ );
3 at time  $t = 1, \dots, m$  do
4   let  $e_t = (p, c)$  where  $p$  is the parent of  $c$ ;
5   if  $y_p^{(t-1)} + y_c^{(t-1)} < 1$  then
6     if  $p$  has two children or ( $p$  gets another child before  $c$  and  $y_c^{(t-1)} = 0$ ) then
7        $y_p^{(t)} \leftarrow 1$ ;
8     end
9     else
10       $y_p^{(t)} \leftarrow \frac{1}{2}, y_c^{(t)} \leftarrow \frac{1}{2}$ 
11    end
12  end
13 end
```

► **Observation 4.** *TWOCHILDRE is correct, that is, $(y_v^{(t)})_{v \in V, t \in [m]}$ satisfies conditions (1) and (2).*

Proof. By construction, the weights $(y_v^{(t)})$ cover the edge e_t at every time step t . Therefore, it only remains to check that no weight is ever decreased. Since no weight is set to 0, it suffices to ensure that no weight is changed from 1 to $\frac{1}{2}$. However, once a vertex has weight 1, all future edges incident to it are already covered, and hence no further updates involving this vertex are performed. ◀

We now proceed with proving that TWOCHILDRE achieves a competitive ratio of $3/2$. As a first step, we establish the following simple properties.

► **Observation 5.** *Fix a time step t and consider the graph $G_t = (V, \{e_1, \dots, e_t\})$ with the vertex weights $(y_v^{(t)})_{v \in V}$ defined by TWOCHILDRE.*

- (a) *If u has no children and $y_u^{(t)} \neq 0$, then $y_u^{(t)} = \frac{1}{2}$, u has no siblings and $y_p^{(t)} = \frac{1}{2}$ as well for the parent p of u .*
- (b) *If a vertex u has exactly one child c and $y_u^{(t)} = 1$, then $y_c^{(t)} = 0$.*

Proof. For part (a), note that the only way a leaf u can receive a non-zero weight is upon arrival of edge (u, p) and the second case of the algorithm is triggered. Thus, at the time t' when (u, p) is revealed, we set $y_u^{(t')} = y_p^{(t')} = \frac{1}{2}$. Moreover, it holds at time t' that u is the only child of p and u gets a child before p gets another child. Since u is a leaf at time t , it follows that, between times $t' + 1$ and t , neither u nor p receives a new child. Therefore, neither y_u nor y_p is updated to 1 before time t , and hence $y_u^{(t)} = y_p^{(t)} = \frac{1}{2}$. Moreover, it still holds at time t that u is the only child of p .

For part (b), if $y_u^{(t)} = 1$, then this weight must have been assigned through execution of the first case upon arrival of edge (u, c) at time t' . Since u has only one child at time $t' \leq t$, the fact that the if-condition in line 6 was triggered implies that u receives another child before c and $y_c^{(t')} = y_c^{(t'-1)} = 0$. Since u still has only one child at time t , no new edge incident to c appears between times $t' + 1$ and t . Therefore, $y_c^{(t)} = 0$. ◀

Now, we have all the prerequisites at hand to prove an upper bound on the competitive ratio of **TWOCHILDRULE**.

► **Theorem 6.** *TWOCHILDRULE is $3/2$ -competitive.*

Proof. Fix a time step t and let G_t be the forest consisting of the edges that appeared up to time t . Let C^* be a minimum vertex cover of G_t , so that $|C^*| = \text{OPT}_t$. Recall that $\text{ALG}_t = \sum_{v \in V} y_v^{(t)}$ and we have to prove that $\text{ALG}_t \leq \frac{3}{2} \text{OPT}_t$.

Our strategy for this is as follows. We equip every $v \in C^*$ with a budget of $3/2$ and redistribute this budget among v and its neighbors. We then show that every $u \in V$ receives at least a budget of $y_u^{(t)}$ from this redistribution. Note that this implies the theorem.

We now describe the redistribution. For each $v \in C^*$, the process is independent of the other vertices. In the following description, any budget that is not explicitly distributed is kept by v .

1. If $y_v^{(t)} = 1$, then v gives $\frac{1}{2}$ to its parent.
2. If $y_v^{(t)} = \frac{1}{2}$, then v gives $\frac{1}{2}$ to its parent and $\frac{1}{2}$ to its first child.
3. If $y_v^{(t)} = 0$, then v gives 1 to its parent and $\frac{1}{2}$ to its first child.

It remains to show that every $u \in V$ receives at least a budget of $y_u^{(t)}$. First, if $u \in C^*$, then u keeps at least a budget of $y_u^{(t)}$ to itself in all three cases. Therefore, assume from now on that $u \notin C^*$. In particular, all neighbors of u belong to C^* . We distinguish three cases: (1) where u has at least two children, (2) where u has no children, and (3) where u has precisely one child.

For case (1), note that, in all three redistribution rules, every vertex gives at least $\frac{1}{2}$ to its parent. Since u has at least two children and all of them belong to C^* , u receives in total at least a budget of 1 from the redistribution.

Next, we consider case (2), i.e., where u has no children. If $y_u^{(t)} = 0$, there is nothing to prove. Otherwise, Observation 5a implies that u has no siblings and $y_u^{(t)} = y_p^{(t)} = \frac{1}{2}$, where p is the parent of u . Since $p \in C^*$, it gives $\frac{1}{2}$ to u , so u is covered.

Last, we consider case (3), i.e., where u has precisely one child c . If $y_u^{(t)} = 1$, then by Observation 5b, we have $y_c^{(t)} = 0$. Since $c \in C^*$, it gives 1 to its parent u , and thus u is covered. If $y_u^{(t)} = \frac{1}{2}$, then $c \in C^*$ gives at least $\frac{1}{2}$ to its parent in all cases, so again u is covered.

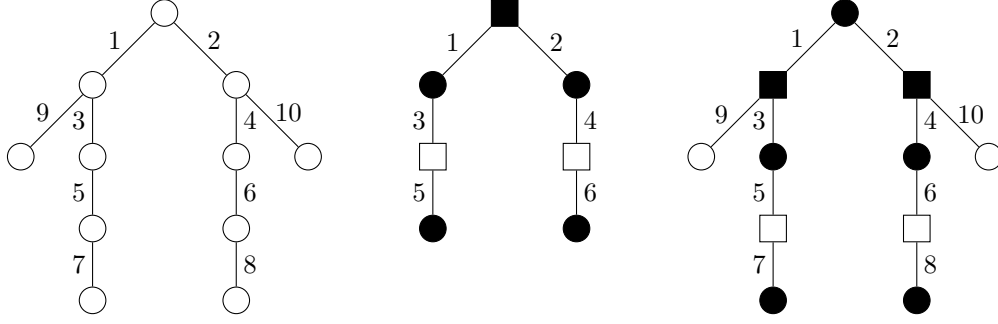
This exhausts all possibilities and therefore completes the proof of the theorem. ◀

It remains open what the competitive ratio of the integral version of the problem is. For this, we formulate the following conjecture, which we verified for all instances on at most 10 vertices via exhaustive search with a computer program.

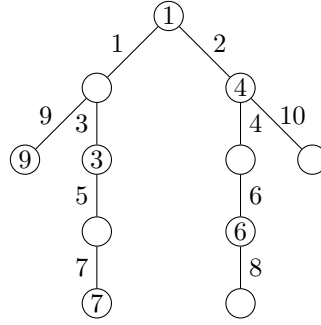
► **Conjecture 7.** *There exists an integral $3/2$ -competitive algorithm for incremental vertex cover on forests.*

In the following, we give an example showing that a simple local rule similar to the one used for the half-integral algorithm cannot achieve the claimed competitive ratio of $3/2$. This suggests that solving the integral version of the problem is substantially more difficult.

► **Proposition 8.** *Consider an integral algorithm for incremental vertex cover on trees. Assume that the tree is rooted and, upon arrival of edge $e = (u, v)$, the decision of the algorithm depends only on the subtree consisting of ancestors and descendants of u and v (and the ordering of its edges). Then the competitive ratio of the algorithm is at least $\frac{5}{3}$.*



■ **Figure 1** Example of a symmetric instance of incremental vertex cover where no symmetric integral solution achieves a competitive ratio of $3/2$. The left subfigure illustrates the instance and the other two subfigures illustrate the two possible symmetric solutions. The filled vertices are the ones chosen by the online algorithm and the square vertices denote the offline optimum at time 6, respectively 10.



■ **Figure 2** An asymmetric integral $3/2$ -competitive solution to the instance presented in Proposition 8. The vertex cover for time t consists of the vertices with a label $\leq t$.

Proof. Consider the example depicted on the left in Figure 1, where the root vertex r is the vertex depicted at the top. Observe that the two subtrees rooted at the two children of r are equivalent in the sense that they result in the same tree and their edges appear in the same order. However, we argue that any integral algorithm that follows the same strategy in both subtrees has competitive ratio at least $\frac{5}{3}$.

At time $t = 2$, the only integral solution with competitive ratio less than 2 is to select the root vertex, so we can assume that the algorithm does so. At times 3 and 4, the algorithm needs to decide whether it selects the parent vertex or child vertex. We assume that the algorithm decides for the same in the two subtrees. If it selects in both the parent vertex (see second subfigure), at times 5 and 6, it needs to select an additional vertex each (in the figure, we have decided wlog. for the child vertex). Then, at time 6, we have $\text{ALG}_6 = 5$, while $\text{OPT}_6 = 3$. Therefore, assume that, at times 3 and 4, the algorithm selects in both subtrees the child vertex (see third subfigure). At times 7 and 8, and at times 9 and 10, an additional vertex must be selected each. Then, at time 10, we have $\text{ALG}_{10} = 7$, while $\text{OPT}_{10} = 4$. To summarize, in either case, the competitive ratio of the algorithm is at least $\frac{5}{3}$. ◀

However, it is not difficult to see that an “asymmetric” $3/2$ -competitive integral solution of the example presented in Proposition 8 exists (see Figure 2).

3.2 A lower bound of 1.5 for fractional incremental vertex cover on trees

In this subsection, we prove that the algorithm `TWOCHILDRE` is optimal by establishing a matching lower bound.

► **Theorem 9.** *Every algorithm for fractional incremental vertex cover on trees has a competitive ratio of at least $3/2$.*

Proof. The instance that we consider is a rooted binary tree of some depth d , which is revealed layer-by-layer. More precisely, *layer i* denotes the set of vertices at distance i from the root and, in *phase i* , we reveal all edges between layer $i - 1$ and i . When we refer to *phase 0*, we mean the time prior to arrival of any edge.

Let `ALG` be an algorithm for the incremental fractional vertex cover problem. For the analysis, we only consider the time steps at the end of a phase. For $i = 1, \dots, d$, let OPT_i denote the weight of an optimal offline vertex cover after phase i and let ALG_i denote the cost of the algorithm after phase i .

We begin by estimating OPT_i . For this, note that equipping the vertices of layers $i - 1, i - 3, i - 5, \dots$ with weight 1 and all other vertices with weight 0 yields a valid fractional vertex cover. For i odd, its total weight is at most

$$\text{OPT}_i \leq 2^{i-1} + \dots + 1 = \sum_{j=0}^{(i-1)/2} 4^j = \frac{4^{(i+1)/2} - 1}{4 - 1} \leq \frac{2^{i+1}}{3}.$$

For i even, its total weight is at most

$$\text{OPT}_i \leq 2^{i-1} + \dots + 2 = 2 \cdot \sum_{j=0}^{(i-2)/2} 4^j = 2 \cdot \frac{4^{i/2} - 1}{4 - 1} \leq \frac{2^{i+1}}{3}.$$

In any case, we have

$$\text{OPT}_i \leq \frac{2^{i+1}}{3}. \quad (1)$$

Next, we analyze the behavior of `ALG`. For this, let α_i be the average weight of vertices in layer i after phase i ($i \in \{0, \dots, d\}$), and let β_i be the average weight of vertices in layer i after phase $i + 1$ ($i \in \{0, \dots, d - 1\}$). Note that, after phase $i + 1$, the weights of the vertices in layer i are not further increased as no new adjacent edges are revealed.

We derive some inequalities that `ALG` must fulfill. First, incrementality ensures that

$$\alpha_i \leq \beta_i \text{ for all } i \in \{0, \dots, d - 1\}. \quad (2)$$

Next, after phase $i + 1$, the edges between layer i and $i + 1$ need to be covered. This implies

$$\beta_i + \alpha_{i+1} \geq 1 \text{ for all } i \in \{0, \dots, d - 1\}. \quad (3)$$

To see this, note that, after phase $i + 1$, the total weight of vertices in layer i is $2^i \beta_i$ and the total weight of vertices in layer $i + 1$ is $2^{i+1} \alpha_{i+1}$. Each of the vertices in layer i contributes its weight to two edges between layer i and $i + 1$ and each of the vertices in layer $i + 1$ contributes its weight to one such edge. Therefore, the edges between layer i and $i + 1$ get a total weight of $2 \cdot 2^i \beta_i + 2^{i+1} \alpha_{i+1}$. Since there are 2^{i+1} such edges, we obtain $\beta_i + \alpha_{i+1} \geq 1$.

Next, we have

$$\text{ALG}_i = 2^i \alpha_i + \sum_{j=0}^{i-1} 2^j \beta_j \text{ for all } i \in \{1, \dots, d\}. \quad (4)$$

158:12 Online and Incremental Fractional Vertex Cover on Trees

Let ρ denote the competitive ratio of ALG. We obtain

$$\rho \geq \frac{\text{ALG}_i}{\text{OPT}_i} \stackrel{(1)(4)}{\geq} \frac{2^i \alpha_i + \sum_{j=0}^{i-1} 2^j \beta_j}{2^{i+1}/3} = \frac{3}{2} \alpha_i + \sum_{j=0}^{i-1} \frac{3}{2^{i+1-j}} \beta_j \text{ for all } i \in \{1, \dots, d\}. \quad (5)$$

Putting inequalities (2), (3), (5) together, we obtain that the competitive ratio of ALG is at least the optimal objective value of the following linear program

$$\begin{aligned} (\text{LP}) \quad & \min \quad \rho \\ & \text{s.t.} \quad \beta_i - \alpha_i \geq 0 \\ & \quad \alpha_{i+1} + \beta_i \geq 1 \\ & \quad \rho - \frac{3}{2} \alpha_{i+1} - \sum_{j=0}^i \frac{3}{2^{i+2-j}} \beta_j \geq 0 \quad (i \in \{0, \dots, d-1\}) \\ & \quad \rho, \alpha_0, \dots, \alpha_d, \beta_0, \dots, \beta_{d-1} \geq 0. \end{aligned}$$

Next, we show a lower bound on the objective value for (LP) (and, hence, a lower bound for the competitive ratio of ALG) by giving a solution for its dual, i.e., using weak duality. For this, we first formulate the dual of (LP). We denote the dual variables for the first constraint by $a_0, \dots, a_{d-1}$, for the second constraint by $b_0, \dots, b_{d-1}$, and for the last constraint by $c_0, \dots, c_{d-1}$. Then the dual is given by

$$\begin{aligned} (\text{D}) \quad & \max \quad \sum_{i=0}^{d-1} b_i \\ & \text{s.t.} \quad -a_0 \leq 0 \\ & \quad -a_i + b_{i-1} - \frac{3}{2} c_{i-1} \leq 0 \quad (i \in \{1, \dots, d\}) \\ & \quad a_i + b_i - \sum_{j=i}^{d-1} \frac{3}{2^{j+2-i}} c_j \leq 0 \quad (i \in \{0, \dots, d-1\}) \\ & \quad \sum_{j=0}^{d-1} c_j \leq 1 \\ & \quad a_i, b_i, c_i \geq 0 \quad (i \in \{0, \dots, d-1\}). \end{aligned}$$

Set $a_0 = \dots = a_{d-1} = 0, b_0 = \dots = b_{d-1} = \frac{3}{2(d+1)}, c_0 = \dots = c_{d-2} = \frac{1}{d+1}, c_{d-1} = \frac{2}{d+1}$. We now show that this is a feasible solution. For the third inequality, we have for $i \in \{0, \dots, d-2\}$

$$\begin{aligned} a_i + b_i - \sum_{j=i}^{d-1} \frac{3}{2^{j+2-i}} c_j &= \frac{3}{2(d+1)} - \sum_{j=i}^{d-2} \frac{3}{2^{j+2-i}} \frac{1}{d+1} - \frac{3}{2^{d+1-i}} \frac{2}{d+1} \\ &= \frac{3}{2(d+1)} \left(1 - \sum_{j=i}^{d-2} \frac{1}{2^{j+1-i}} - \frac{2}{2^{d-i}} \right) \\ &= \frac{3}{2(d+1)} \left(1 - \sum_{j=1}^{d-1-i} \frac{1}{2^j} - \frac{1}{2^{d-1-i}} \right) = 0. \end{aligned}$$

For $i = d - 1$, we have in the third inequality

$$a_i + b_i - \sum_{j=i}^{d-1} \frac{3}{2^{j+2-i}} c_j = \frac{3}{2(d+1)} - \frac{3}{2^2} \frac{2}{d+1} = 0.$$

Hence, the third inequality is fulfilled. For all other inequalities, it is immediate that they are fulfilled.

Therefore, the described solution is feasible and its objective value is

$$\sum_{i=0}^{d-1} b_i = \frac{3d}{2(d+1)}.$$

Therefore, we obtain that the competitive ratio of ALG on a binary tree of depth d is at least $\frac{3d}{2(d+1)}$. Letting $d \rightarrow \infty$ yields the statement of the Theorem. $\blacktriangleleft$

► **Remark 10.** It is worth noting that our lower bound construction also applies to the *vertex arrival model*, where an ordering of the vertices $V = \{v_1, \dots, v_n\}$ is given and, in time step t , all edges between v_t and $v_1, \dots, v_{t-1}$ are revealed. Furthermore, our construction also applies to trees (not only forests), meaning that, at every time step, the graph induced by the revealed edges is connected.

References

- 1 Aaron Bernstein, Yann Disser, Martin Groß, and Sandra Himburg. General bounds for incremental maximization. *Mathematical Programming*, 191(2):953–979, 2022. doi:10.1007/s10107-020-01576-0.
- 2 Avrim Blum, Prasad Chalasani, Don Coppersmith, William R. Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. The minimum latency problem. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC)*, pages 163–171, 1994. doi:10.1145/195058.195125.
- 3 Niv Buchbinder, Kamal Jain, and Joseph (Seffi) Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of the 15th Annual European Symposium on Algorithms (ESA)*, pages 253–264, 2007. doi:10.1007/978-3-540-75520-3_24.
- 4 Niv Buchbinder, Danny Segev, and Yevgeny Tkach. Online algorithms for maximum cardinality matching with edge arrivals. *Algorithmica*, 81(5):1781–1799, 2019. doi:10.1007/s00453-018-0505-7.
- 5 Marek Chrobak, Claire Kenyon, John Noga, and Neal E. Young. Incremental medians via online bidding. *Algorithmica*, 50(4):455–478, 2008. doi:10.1007/s00453-007-9005-x.
- 6 Ilan Reuven Cohen and David Wajc. Randomized online matching in regular graphs. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 960–979, 2018. doi:10.1137/1.9781611975031.62.
- 7 Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 101–107, 2013. doi:10.1137/1.9781611973105.7.
- 8 Yann Disser, Max Klimm, Annette Lutz, and David Weckbecker. Fractionally subadditive maximization under an incremental knapsack constraint with applications to incremental flows. *SIAM Journal on Discrete Mathematics*, 38(1):764–789, 2024. doi:10.1137/23M1569265.
- 9 Yann Disser, Max Klimm, Nicole Megow, and Sebastian Stiller. Packing a knapsack of unknown capacity. *SIAM Journal on Discrete Mathematics*, 31(3):1477–1497, 2017. doi:10.1137/16M1070049.

- 10 Yann Disser, Max Klimm, Kevin Schewior, and David Weckbecker. Incremental maximization via continuization. In *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 47:1–47:17, 2023. doi:10.4230/LIPIcs.ICALP.2023.47.
- 11 Yann Disser and David Weckbecker. Incremental maximization for a broad class of objectives. In *Proceedings of the 33rd Annual European Symposium on Algorithms (ESA)*, pages 92:1–92:13, 2025. doi:10.4230/LIPIcs.ESA.2025.92.
- 12 Ryo Fujita, Yusuke Kobayashi, and Kazuhisa Makino. Robust matchings and matroid intersections. *SIAM Journal on Discrete Mathematics*, 27(3):1234–1256, 2013. doi:10.1137/100808800.
- 13 Buddhima Gamlath, Michael Kapralov, Andreas Maggiori, Ola Svensson, and David Wajc. Online matching with general arrivals. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 26–37, 2019. doi:10.1109/FOCS.2019.00011.
- 14 Michel X. Goemans and Jon M. Kleinberg. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, 82:111–124, 1998. doi:10.1007/BF01585867.
- 15 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.
- 16 Refael Hassin and Shlomi Rubinstein. Robust matchings. *SIAM Journal on Discrete Mathematics*, 15(4):530–537, 2002. doi:10.1137/S0895480198332156.
- 17 Tianle Jiang and Yuhao Zhang. Edge arrival online matching: The power of free disposal on acyclic graphs. In *Web and Internet Economics*, pages 591–608, 2026. doi:10.1007/978-3-032-08560-3_33.
- 18 Naonori Kakimura and Kazuhisa Makino. Robust independence systems. *SIAM Journal on Discrete Mathematics*, 27(3):1257–1273, 2013. doi:10.1137/120899480.
- 19 Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 352–358, 1990. doi:10.1145/100216.100262.
- 20 Euiwoong Lee and Sahil Singla. Maximum matching in the online batch-arrival model. *ACM Transactions on Algorithms*, 16(4):49:1–49:31, 2020. doi:10.1145/3399676.
- 21 Guolong Lin, Chandrashekhar Nagarajan, Rajmohan Rajaraman, and David P. Williamson. A general approach for incremental approximation and hierarchical clustering. *SIAM Journal on Computing*, 39(8):3633–3669, 2010. doi:10.1137/070698257.
- 22 Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: An approach based on strongly factor-revealing lps. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 597–606, 2011. doi:10.1145/1993636.1993716.
- 23 Jannik Matuschke, Martin Skutella, and José A. Soto. Robust randomized matchings. *Mathematics of Operations Research*, 43(2):675–692, 2018. doi:10.1287/moor.2017.0878.
- 24 Nicole Megow and Julián Mestre. Instance-sensitive robustness guarantees for sequencing with unknown packing and covering constraints. In *Proceedings of the 4th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 495–504, 2013. doi:10.1145/2422436.2422490.
- 25 Ramgopal R. Mettu and C. Greg Plaxton. The online median problem. *SIAM Journal on Computing*, 32(3):816–832, 2003. doi:10.1137/S0097539701383443.
- 26 Joseph (Seffi) Naor and David Wajc. Near-optimum online ad allocation for targeted advertising. *ACM Transactions on Economics and Computation*, 6(3–4):16:1–16:20, 2018. doi:10.1145/3105447.
- 27 C. Greg Plaxton. Approximation algorithms for hierarchical location problems. *Journal of Computer and System Sciences*, 72(3):425–443, 2006. doi:10.1016/j.jcss.2005.09.004.
- 28 Zhihao Gavin Tang. Optimal competitive ratio of two-sided online bipartite matching, 2026. doi:10.48550/arXiv.2602.18049.

- 29 Zhihao Gavin Tang and Yuhao Zhang. Improved bounds for fractional online matching problems. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pages 279–307, 2024. doi:10.1145/3670865.3673459.
- 30 Yajun Wang and Sam Chiu-wai Wong. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 1070–1081, 2015. doi:10.1007/978-3-662-47672-7_87.