Algorithmica (2022) 84:124–149
https://doi.org/10.1007/s00453-021-00886-9

# Approximating Dynamic Weighted Vertex Cover with Soft Capacities

Hao-Ting Wei[1] · Wing-Kai Hon[2] · Paul Horn[3] · Chung-Shou Liao[1] · Kunihiko Sadakane[4]

Received: 10 November 2020 / Accepted: 10 October 2021 / Published online: 16 November 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

This study considers the *soft* capacitated vertex cover problem in a dynamic setting. This problem generalizes the dynamic model of the vertex cover problem, which has been intensively studied in recent years. Given a dynamically changing vertex-weighted graph $G = (V, E)$, which allows edge insertions and edge deletions, the goal is to design a data structure that maintains an approximate minimum vertex cover while satisfying the capacity constraint of each vertex. That is, when picking a copy of a vertex $v$ in the cover, the number of $v$'s incident edges covered by the copy is up to a given capacity of $v$. We extend Bhattacharya et al.'s work [SODA'15 and ICALP'15] to obtain a deterministic primal-dual algorithm for maintaining a constant-factor approximate minimum capacitated vertex cover with $O(\log n/\epsilon)$ amortized

This work was partially supported by NSA Young Investigator Grant H98230-15-1-0258, Simons Collaboration Grant #525039, and MOST Taiwan Grants 105-2628-E-007-010-MY3 and 109-2634-F-007-018.

✉ Chung-Shou Liao
csliao@ie.nthu.edu.tw

Hao-Ting Wei
s104034526@m104.nthu.edu.tw

Wing-Kai Hon
wkhon@cs.nthu.edu.tw

Paul Horn
paul.horn@du.edu

Kunihiko Sadakane
sada@mist.i.u-tokyo.ac.jp

[1] Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan

[2] Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan

[3] Department of Mathematics, University of Denver, Denver, USA

[4] Department of Mathematical Informatics, The University of Tokyo, Tokyo, Japan

update time, where $n$ is the number of vertices in the graph. The algorithm can be extended to (1) a more general model in which each edge is associated with a non-uniform and unsplittable demand, and (2) the more general capacitated set cover problem.

**Keywords** Dynamic algorithm · Vertex cover · Approximation algorithm

## 1 Introduction

Dynamic algorithms have received fast-growing attention in the past decades, especially for some classical combinatorial optimization problems such as connectivity [1,10,13,15], routing [2,8,21,22], vertex cover and maximum matching [3–6,16–20]. This paper focuses on the fully dynamic model for the vertex cover problem, which has been intensively studied in recent years. Given a vertex-weighted graph $G = (V, E)$, here we called the weight on each vertex as cost to avoid abusing this term in the following section. Such a graph is constantly updated due to a sequence of edge insertions and edge deletions, where the objective is to maintain a subset of vertices $S \subseteq V$ at any given time, such that every edge is incident to at least one vertex in $S$ and the total cost of $S$ is minimized. We consider a generalization of the problem, where each vertex is associated with a given capacity. When picking a copy of a vertex $v$ in the multiset $S$, the number of its incident edges that can be covered by such a copy is bounded by $v$'s given capacity. The objective is to find a *soft* capacitated weighted vertex cover $S$ with minimum cost, i.e. $\sum_{v \in S} c_v x_v$ is minimized, as well as an assignment of edges such that the number of edges assigned to a vertex $v$ in $S$ is at most $k_v x_v$, where $c_v$ is the cost of $v$, $k_v$ is the capacity of $v$, and $x_v$ is the number of selected copies of $v$ in $S$. Assume there is no bound on $x_v$. The static model of this generalization is the so-called *soft capacitated vertex cover* problem, introduced by Guha et al. [11], whereas if each $x_v$ is associated with a bound, it is called the *hard capacitated vertex cover* problem, introduced by Chuzhoy and Naor [9].

*Prior work* For the vertex cover problem in a dynamic setting, Ivkovic and Lloyd [14] presented the pioneering work wherein their fully dynamic algorithm maintains a 2-approximation factor to vertex cover with $O((n+m)^{0.7072})$ update time, where $n$ is the number of vertices and $m$ is the number of edges. Onak and Rubinfeld [17] designed a randomized data structure that maintains a large constant approximation ratio with $O(\log^2 n)$ amortized update time in expectation; this is the first result that achieves a constant approximation factor with polylogarithmic update time. Baswana et al. [3] designed another randomized data structure which improves the approximation ratio to two, and simultaneously improved the amortized update time to $O(\log n)$. Recently, Solomon [20] gave the currently best randomized algorithm, which maintains a 2-approximate vertex cover with $O(1)$ amortized update time.

For deterministic data structures, Onak and Rubinfeld [17] presented a data structure that maintains an $O(\log n)$-approximation algorithm with $O(\log^2 n)$ amortized update time. Bhattacharya et al. [6] proposed the first deterministic data structure that maintains a constant ratio, precisely, a $(2 + \epsilon)$-approximation to vertex cover with polylogarithmic $O(\log n/\epsilon^2)$ amortized updated time. Existing work also considered the worst-case update time. Neiman and Solomon [16] provided a 2-approximation dynamic algorithm with $O(\sqrt{m})$ worst-case update time. Peleg and Solomon [18] improved the worst-case update time to $O(\gamma/\epsilon^2)$, where $\gamma$ is the arboricity of the input graph. Very recently, Bhattacharya et al. [4] extended their hierarchical data structure to achieve the currently best worst-case update time of $O(poly(1/\epsilon)\cdot\log^3 n)$[1]. Note that the above studies only discussed the unweighted vertex cover problem, the objective of which is to find a vertex cover with minimum cardinality.

Consider the dynamic (weighted) set cover problem. In the statics setting, given a universe $U$ of $n$ elements and a family $F$ of $m$ sets. Each set is associated with non-negative costs. The objective is to find a subfamily of sets $S \in F$ of minimum cost that covers $U$. The dynamic set cover problem follows a sequence of element insertions and deletions, the objective is to maintain a subfamily of sets $S \in F$ at any given time, such that the union of $S$ covers the elements that we seen so far.

Bhattacharya et al. [7] used a hierarchical data structure similar to that reported in [6], and and presented an algorithm with $O(f^2)$-approximation ratio and $O(f\log(n+m))$ amortized updated time, where $f$ is the maximum frequency of an element. Very recently, Gupta et al. [12] improved the amortized update time to $O(f^2)$, albeit the dynamic algorithm achieves a higher approximation ratio of $O(f^3)$. They also offered another $O(\log n)$-approximation dynamic algorithm in $O(f\log n)$ amortized update time. Bhattacharya et al. [5] simultaneously derived the same outcome with $O(f^3)$-approximation ratio and $O(f^2)$ amortized update time for the unweighted set cover problem. Table 1 presents a summary of the above results.

*Our Contribution* In this study we investigate the *soft* capacitated vertex cover problem in the dynamic setting, where there is no bound on the number of copies of each vertex that can be selected. We refer to the primal-dual technique reported in [11], and present the first deterministic algorithm for this problem, which can maintain an $O(1)$-approximate minimum capacitated (weighted) vertex cover with $O(\log n/\epsilon)$ amortized update time. The algorithm can be extended to a more general model in which each edge is associated with a given demand, and the demand has to be assigned to an incident vertex. That is, the demand of each edge is *non-uniform* and *unsplittable*. Also, it can be extended to solve the more general capacitated set cover problem, where the input graph is a hyper-graph, and each edge may connect to multiple vertices.

The proposed dynamic mechanism builds on Bhattacharya et al.'s $(\alpha, \beta)$-partition structure [6,7], but a *careful adaptation* has to be made to cope with the newly introduced capacity constraint. In dynamic vertex cover [6], the authors considered dynamically maintaining the dual problem of the relaxed vertex cover, that is, the fractional matching problem. They partitioned all of the vertices into different levels and designed a *weight* function for the fractional matching problem.

Briefly, applying the *fractional matching* technique in Bhattacharya et al.'s algorithm cannot directly lead to a constant approximation ratio in the capacitated vertex cover problem. The crux of our result is the re-design of a key parameter, *weight* function of a vertex, in the dual model. Details are shown in the next section.

[1] We thank the original author of [4] for clarifying the dependency of $\epsilon$ in the update time.

**Table 1** Summary of results for unweighted (resp. weighted) minimum vertex cover (UMVC (resp. WMVC)), unweighted (resp. weighted) minimum set cover (UMSC (resp. WMSC)), where $f$ is the maximum frequency of an element, and weighted minimum capacitated vertex (resp. set) cover (WMCVC (resp. WMCSC))

| Problem | Approx. Guarantee | Update time | Data Structure | References |
|---|---|---|---|---|
| UMVC | $O(1)$ | $O(\log^2 n)$ amortized | Randomized | STOC'10 [17] |
| UMVC | 2 | $O(\log n)$ amortized | Randomized | SICOMP'15 [3] |
| UMVC | 2 | $O(1)$ amortized | Randomized | FOCS'16 [20] |
| UMVC | 2 | $O(\sqrt{m})$ worst-case | Deterministic | TALG'16 [16] |
| UMVC | $2+\epsilon$ | $O(\log n/\epsilon^2)$ amortized | Deterministic | SODA'15 [6] |
| UMVC | $2+\epsilon$ | $O(\gamma/\epsilon^2)$ worst-case | Deterministic | SODA'16 [18] |
| UMVC | $2+\epsilon$ | $O(poly(1/\epsilon) \cdot \log^3 n)$ worst-case | Deterministic | SODA'17 [4] |
| UMSC | $O(f^3)$ | $O(f^2)$ amortized | Deterministic | IPCO'17 [5] |
| WMSC | $O(f^2)$ | $O(f \log(n+m))$ amortized | Deterministic | ICALP'15 [7] |
| WMSC | $O(f^3)$ | $O(f^2)$ amortized | Deterministic | STOC'17 [12] |
|  | $O(\log n)$ | $O(f \log n)$ amortized |  |  |
| WMCVC | $O(1)$ | $O(\log n/\epsilon)$ amortized | Deterministic | This paper |
| WMCSC | $O(f^2)$ | $O(f \log(n+m))$ amortized | Deterministic | This paper |

### 1.1 Overview of Our Technique

First, we recall the mathematical model of the capacitated vertex cover problem which was first introduced by Guha et al. [11]. In this model, $y_{ev}$ serves as a binary variable that indicates whether an edge $e$ is covered by a vertex $v$. Let $N_v$ be the set of incident edges of $v$, $k_v$ and $c_v$ be the capacity and the cost of a vertex $v$, respectively. Let $x_v$ be the number of selected copies of a vertex $v$. An integer program (IP) model of the problem can be formulated as follows (the minimization program on the left):

$$\begin{array}{lll} \textbf{Min} & \sum_v c_v x_v & \\ \textbf{s.t} & y_{ev} + y_{eu} \geq 1, & \forall e = (u, v) \in E \\ & k_v x_v - \sum_{e \in N_v} y_{ev} \geq 0, & \forall v \in V \\ & x_v \geq y_{ev}, & \forall v \in e, \forall e \in E \\ & y_{ev} \in \{0, 1\}, & \forall v \in e, \forall e \in E \\ & x_v \in \mathbb{N}, & \forall v \in V \end{array}$$

$$\begin{array}{lll} \textbf{Max} & \sum_{e \in E} \pi_e & \\ \textbf{s.t} & k_v q_v + \sum_{e \in N_v} l_{ev} \leq c_v, & \forall v \in V \\ & q_v + l_{ev} \geq \pi_e, & \forall v \in e, \forall e \in E \\ & q_v \geq 0, & \forall v \in V \\ & l_{ev} \geq 0, & \forall v \in e, \forall e \in E \\ & \pi_e \geq 0, & \forall e \in E \end{array}$$

If we allow a relaxation of the above primal form, i.e., dropping the integrality constraints, its dual problem yields a maximization problem. The linear program for the dual can be formulated as shown in the above (the maximization program on the right; also see [11]). One may consider this as a variant of the *packing* problem, where we want to pack a value of $\pi_e$ for each edge $e$, so that the sum of the packed values is maximized. Packing of $e$ is limited by the sum of $q_v$ and $l_{ev}$, where $q_v$ is the *global* ability of a vertex $v$ emitted to $v$'s incident edges, and $l_{ev}$ is the *local* ability of $v$ distributed to its incident edge $e$.

In this study, we incorporate the above IP model with its LP relaxation for capacitated vertex cover into the dynamic mechanism proposed Bhattacharya et al.'s [6,7]. They devised the *weight* function of a vertex (in the dual model). Based on the *weight* function, we can select a subset of vertices that obtains a feasible vertex cover. They also allowed a flexible range for *weight* function to quickly adjust the solution for dynamic updates while preserving its approximation quality. Due to the additional capacity constraint in our problem, a new *weight* function is obviously required.

*Technical Challenges* There are two major differences between our algorithm and Bhattacharya et al.'s [6,7]. First, the capacity constraint in the primal problem leads to the two variables $q_v$ and $l_{ev}$ in the dual problem in which we have to balance their values when approaching $c_v$ to maximize the dual objective. By contrast, the previous work considered one dual variable $l_{ev}$ without the restriction on the *coverage* of a vertex. We thus re-design *weight* function of a vertex to specifically consider the capacitated scenario. Yet, even with the new definition of *weight* function, there is still a second challenge on how to approximate the solution within a constant factor in the dynamic environment. In order to achieve polylogarithmic amortized update time, Bhattacharya et al.'s *fractional matching* approach assigns the value of all $v$'s incident edges to $v$, which, however, may result in a non-constant $h$, hidden in the approximation ratio, where $h$ is the largest number of copies selected in the cover. We observe that we cannot remove $h$ from the approximation guarantee based on the $(\alpha, \beta)$-partition structure if we just select the minimum value of $\alpha$, as it is done in (Bhattacharya et al. [6,7]). The key insight is that we show a bound on the value of $\alpha$, which restricts the updates of the dynamic mechanism. With the help of this insight, we are able to revise the setting of $\alpha$ to derive a constant approximation ratio, while maintaining the $O(\log n/\epsilon)$ update time.

## 2 Level Scheme and Its Key Property

The core of Bhattacharya et al.'s $(\alpha, \beta)$-partition structure [6,7]) is a *level scheme* [17] that is used to maintain a feasible solution in their dual problem. In this section, we demonstrate (in a different way from the original papers) how this scheme can be applied to our dual problem, and describe the key property that the scheme guarantees.

A *level scheme* is an assignment $\ell : V \rightarrow \{0, 1, \ldots, L\}$ such that every vertex $v \in V$ has a level $\ell(v)$. Let $c_{\min}$ and $c_{\max}$ denote the minimum and maximum costs of a vertex, respectively. For our case, we set $L = \lceil \log_\beta(n\mu\alpha/c_{\min}) \rceil$ for some $\alpha, \beta > 1$ and $\mu > c_{\max}$. Based on $\ell$, each edge $(u, v)$ is also associated with a level $\ell(u, v)$, where $\ell(u, v) = \max\{\ell(u), \ell(v)\}$. An edge is assigned to the higher-level endpoint, and ties are broken arbitrarily if both endpoints have the same level. Conceptually, based on the level status of the vertices, the *level scheme* transforms an undirected graph into a directed structure. Each edge $(u, v)$ has a weight $w(u, v)$ according to its level, such that $w(u, v) = \mu\beta^{-\ell(u,v)}$. Each vertex $v$ also has a weight $W_v$, which is defined based on the incident edges of $v$ and their corresponding levels. Before giving details on $W_v$, we first define some notations. Let $N_v = \{u \mid (u, v) \in E\}$ be the set of vertices adjacent to $v$ (i.e., the neighbors of $v$). Let $N_v(i)$ denote the set of level-$i$ neighbors of $v$, and $N_v(i, j)$ denote the set of $v$'s neighbors whose levels are in the range $[i, j]$. That is, $N_v(i) = \{u \mid (u, v) \in E \wedge \ell(u) = i\}$ and $N_v(i, j) = \{u \mid (u, v) \in E \wedge \ell(u) \in [i, j]\}$. The *degree* of a vertex $v$ is denoted by $D_v = |N_v|$. Similarly, we define $D_v(i) = |N_v(i)|$ and $D_v(i, j) = |N_v(i, j)|$. Finally, we use $\delta(v)$ to denote the set of edges assigned to a vertex $v$. Now, the weight $W_v$ of a vertex $v$ is defined as follows:

- **Case** 1 $D_v(0, \ell(v)) > k_v$:

$$W_v = k_v\mu\beta^{-\ell(v)} + \sum_{i>\ell(v)} \min\{k_v, D_v(i)\}\mu\beta^{-i}$$

- **Case** 2 $D_v(0, \ell(v)) \le k_v$:

$$W_v = D_v(0, \ell(v))\mu\beta^{-\ell(v)} + \sum_{i>\ell(v)} \min\{k_v, D_v(i)\}\mu\beta^{-i}$$

The intuition behind this weight function refers to the primal-dual algorithm proposed by Guha et al. [11], where by the capacity constraint, in each iteration, a vertex can increase its dual value by at most $k_v$ edges, no matter how many incident edges it has. By a similar idea, we consider whether the number of level-$i$ neighbors of $v$, $0 \le i \le \ell(v)$, is larger than the capacity of $v$, and use it to define the weight of a vertex $v$. Note that the total weight of the edges that are assigned to $v$ or incident to $v$ can contribute at most

$k_v w(u,v)$ to $W_v$, which is different from the [6,7]. (In Bhattacharya et al. [6,7], since they do not need to consider the capacity constraint, the weight of a vertex is simply the sum of the weights of all its incident edges; that is, $W_v = \sum_{u \in N(v)} w(u,v)$.) Moreover, our proposed weight function provides a relationship between the weights of different levels, which plays an essential role when we consider the dynamic setting. Briefly, the weight of a vertex has two components: one that is dependent on the incident edges with level $\ell(v)$, and the other that is dependent on the remaining incident edges. For convenience, we call the former component *Internal*$_v$ and the latter component as *External*$_v$. Moreover, we have:

$$\mathit{External}_v \quad \leq \quad k_v \sum_{i > \ell(v)} \mu\beta^{-i} \quad \leq \quad (1/(\beta-1))k_v\mu\beta^{-\ell(v)}.$$

In general, an arbitrary level scheme cannot be used to solve our problem. What we need is a *valid* level scheme, which is defined as follows.

**Definition 1** A level scheme is *valid* if $W_v \leq c_v$, for every vertex $v$.

**Lemma 1** *Let $V_0$ denote the set of level-0 vertices in a valid level scheme. Then, $V \setminus V_0$ forms a* vertex cover *of $G$.*

***Proof*** Consider any edge $(u,v) \in E$. We claim that at least one of its endpoints must be in $V \setminus V_0$. Suppose that the claim is false which implies that $\ell(u) = \ell(v) = 0$ and $w(u,v) = \mu > c_{\max}$. Since $w(u,v)$ appears in *Internal*$_v$, we have $W_v \geq w(u,v)$. As a result, $c_v \geq W_v \geq \mu > c_{\max}$, which leads to a contradiction. The claim thus follows, and so does the lemma.

□

The above lemma implies that no edge is assigned to any level-0 vertex. In our mechanism, we will maintain a valid level scheme, based on which each vertex in $V \setminus V_0$ picks enough copies to cover all the edges assigned to it; this forms a valid capacitated vertex cover.

Next, we define the notion of *tightness*, which is used to measure how good a valid level scheme performs.

**Definition 2** A valid level scheme with an associated edge assignment is $\varepsilon$-*tight* if for every vertex $v$ with $|\delta(v)| > 0$, $W_v \in (c_v/\varepsilon, c_v]$.

**Lemma 2** *Given an $\varepsilon$-tight valid level scheme, we can obtain an $\varepsilon(2(\beta/(\beta-1))+1)$-approximate solution to the* weighted minimum capacitated vertex cover (WMCVC) *problem.*

***Proof*** First, we fix an arbitrary edge assignment that is consistent with the given valid level scheme. For each vertex $v$ with $|\delta(v)| > 0$, we pick $\lceil |\delta(v)|/k_v \rceil$ copies to cover all the $|\delta(v)|$ edges assigned to it. To analyze the total cost of this capacitated vertex cover, we relate it to the value $\sum_e \pi_e$ of a certain feasible solution of the dual problem, whose corresponding values of $q_v$ and $l_{ev}$ are as follows:

For every vertex $v$:

- if $\lceil|\delta(v)|/k_v\rceil > 1$: $q_v = \mu\beta^{-\ell(v)}$, and $l_{ev} = 0$;
- if $\lceil|\delta(v)|/k_v\rceil \le 1$: $q_v = \mu\sum_{i|D_v(i)>k_v}\beta^{-i}$,

$l_{ev} = 0$ if $D_v(\ell(e)) > k_v$, and $l_{ev} = \mu\beta^{-\ell(e)}$ otherwise.
For every edge $e$: $\pi_e = \mu\beta^{-\ell(e)}$.

It is easy to verify that the above choices of $q_v$, $l_{ev}$, and $\pi_e$ give a feasible solution to the dual problem.

For the total cost of our solution, we separate the analysis into two parts, based on the multiplicity of the vertex:

- **Case** 1 $\lceil|\delta(v)|/k_v\rceil > 1$: In this case, the external component of $W_v$ is at most $1/(\beta-1)$ of the internal component, so $W_v \le (\beta/(\beta-1))k_v q_v$. Then, the cost of all copies of $v$ is:

$$\begin{aligned}\lceil|\delta(v)|/k_v\rceil\cdot c_v &\le \lceil|\delta(v)|/k_v\rceil\cdot\varepsilon\cdot W_v\\ &\le 2\cdot\frac{|\delta(v)|}{k_v}\cdot\varepsilon\cdot(\beta/(\beta-1))k_v q_v \;=\; 2\varepsilon(\beta/(\beta-1))\cdot\sum_{e\in\delta(v)}\pi_e.\end{aligned}$$

- **Case** 2 $\lceil|\delta(v)|/k_v\rceil = 1$: In this case, we pick one copy of vertex $v$, whose cost is:

$$c_v \le \varepsilon\cdot W_v \;\le \varepsilon\cdot\sum_{e\sim v}\pi_e \;=\; \varepsilon\cdot\left(\sum_{e\in\delta(v)}\pi_e + \sum_{e\notin\delta(v),\,e\sim v}\pi_e\right),$$

where $e\sim v$ denotes $e$ is an edge incident to $v$.

In summary, the total cost is bounded by

$$\begin{aligned}&\sum_v\left(\max\{\varepsilon, 2\varepsilon(\beta/(\beta-1))\}\sum_{e\in\delta(v)}\pi_e + \varepsilon\sum_{e\notin\delta(v),\,e\sim v}\pi_e\right)\\ &=\sum_v\left(2\varepsilon(\beta/(\beta-1))\sum_{e\in\delta(v)}\pi_e + \varepsilon\sum_{e\notin\delta(v),\,e\sim v}\pi_e\right)\\ &=\varepsilon(2(\beta/(\beta-1))+1)\sum_e\pi_e\\ &\le\varepsilon(2(\beta/(\beta-1))+1)\cdot OPT,\end{aligned}$$

where $OPT$ denotes the optimal solution of the dual problem, which is also a lower bound of the cost of any weighted capacitated vertex cover. □

The next section discusses how to dynamically maintain an $\varepsilon$-tight level scheme, for some constant factor $\varepsilon$ and with amortized $O(\log n/\epsilon)$ update time. Before that, we show a greedy approach to get a $(\beta+1)$-tight level scheme to the static problem as a warm up.

First, we have the following definition.

**Definition 3** A valid level scheme λ is *improvable* if some vertex can drop its level to get another level scheme λ′ such that λ′ is valid; otherwise, we say λ is *non-improvable*.

**Lemma 3** *If a valid level scheme λ is non-improvable, then λ is* $(\beta + 1)$*-tight.*

***Proof*** To prove this lemma, we compare the weight $W_v$ of a vertex $v$ when its level is set to $i$ and $i + 1$, respectively (while the level of every other vertex remains unchanged).

- **Case** 1 $D_v(0, i) < k_v$ and $D_v(0, i + 1) < k_v$:

$$\begin{aligned} W_v(i + 1) &= D_v(0, i + 1)\mu\beta^{-(i+1)} + \sum_{j>i+1} \min\{k_v, D_v(j)\}\mu\beta^{-j}; \\ W_v(i) &= D_v(0, i)\mu\beta^{-(i)} + D_v(i + 1)\mu\beta^{-(i+1)} + \sum_{j>i+1} \min\{k_v, D_v(j)\}\mu\beta^{-j} \\ &\le \beta W_v(i + 1); \end{aligned}$$

- **Case** 2 $D_v(0, i + 1) \ge k_v$:

$$\begin{aligned} W_v(i + 1) &= k_v\mu\beta^{-(i+1)} + \sum_{j>i+1} \min\{k_v, D_v(j)\}\mu\beta^{-j}; \\ W_v(i) &= \min\{D_v(0, i), k_v\}\mu\beta^{-(i)} + \min\{k_v, D_v(i + 1)\}\mu\beta^{-(i+1)} \\ &\quad + \sum_{j>i+1} \min\{k_v, D_v(j)\}\mu\beta^{-j} \\ &\le (\beta + 1)W_v(i + 1); \end{aligned}$$

In both cases, the weight $W_v(i)$ is at most $(\beta + 1)$ times of $W_v(i + 1)$. Thus, if a vertex cannot drop its level, either its current level is 0, or by doing so we have $W_v(\ell(v) - 1) > c_v$; the latter implies that the current value of $W_v = W_v(\ell(v))$ is larger than $c_v/(\beta + 1)$. Thus, if no vertex can drop its level, then the level scheme is $(\beta + 1)$-tight. □

If we set the level of every vertex to $L$ initially, it is easy to check that by our choice of $L$ as $\lceil \log_\beta(n\mu\alpha/c_{\min})\rceil$, such a level scheme is valid. Next, we examine each vertex one by one, and drop its level as much as possible while the scheme remains valid. In the end, we will obtain a non-improvable scheme, so that by the above lemma, the scheme is $(\beta + 1)$-tight. This implies a $(\beta + 1)(2(\beta/(\beta - 1)) + 1)$-approximate solution for the WMCVC problem.

## 3 Maintaining an $\alpha(\beta + 1)$-tight Level Scheme Dynamically

In this section, we present our $O(1)$-approximation algorithm for the WMCVC problem, with amortized $O(\log n)$ update time for each edge insertion and edge deletion. We first state an invariant that is maintained throughout by our algorithm, and show how the latter is done. Next, we analyze the time required to maintain the invariant with the potential method, and show that our proposed method can be updated efficiently as desired. Similar to [6,7], we allow a flexible range for the weight of a vertex, $W_v$, by a multiplicative constant $\alpha$ so that we can obtain an $O(\log n)$ amortized update time. See Fig. 1 for an illustration of the overall framework.

Let $c^*_v$ be $c_v/\alpha(\beta + 1)$. The invariant that we maintain is as follows.

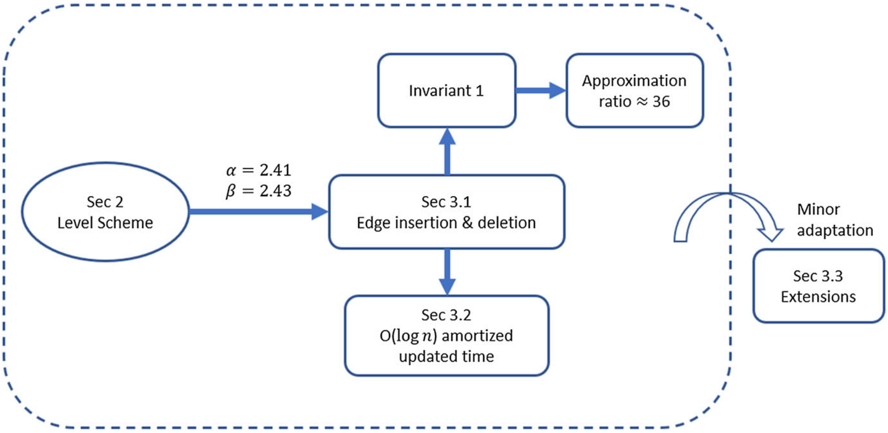

**Fig. 1** Framework of Sect. 3

**Invariant 1** *(1) For every vertex $v \in V \setminus V_0$, it holds that $c_v^* \leq W_v \leq c_v$, and (2) for every vertex $v \in V_0$, it holds that $W_v \leq c_v$ .*

By maintaining the above invariant, we will automatically obtain an $\alpha(\beta+1)$-tight valid scheme. As mentioned, we will choose a value for $\alpha$ in order to remove $h$ from the approximation ratio, where h is the largest number of copies selected in the cover. In particular, we will set $\alpha = (2\beta+1)/\beta + 2\epsilon$, where $0 < \epsilon < 1$ to balance the update time, and $\beta = 2.43$ to minimize the approximation ratio, so that we achieve the following theorem.

**Theorem 2** *There exists a dynamic level scheme $\lambda$ which can achieve a constant approximation ratio ($\approx$ 36) for the WMCVC problem with $O(\log n/\epsilon)$ amortized update time.*

The remainder of this section is devoted to proving Theorem 2.

### 3.1 The Algorithm: Handling Insertion or Deletion of an Edge

We now show how to maintain the invariant under edge insertions and deletions. A vertex is called *dirty* if it violates Invariant 1, and *clean* otherwise. Initially, the graph is empty, so that every vertex is clean and is at level zero.

Assume that at the time instant just prior to the $t^{th}$ update, all vertices are clean. When the $t^{th}$ update takes place, which either inserts or deletes an edge $e = (u, v)$, we need to adjust the weights of $u$ and $v$ accordingly. Due to this adjustment, the vertices $u$, or $v$, or both may become dirty. To recover from this, we call the procedure Fix. The pseudo code of the update algorithm (Algorithm 1) appears in the next page.

The proposed algorithm ensures that Invariant 1 is maintained after each update, so that the dynamic scheme is $\alpha(\beta+1)$-tight as desired. To complete the discussion, as well as the proof of Theorem 2, it remains to show that each update can be performed efficiently, in amortized $O(\log n)$ time.

**Algorithm 1**

```
if an edge $e = (u, v)$ has been inserted then
    Set $\ell(e) = \max\{\ell(u), \ell(v)\}$ and set $w(u, v) = \mu\beta^{-\ell(e)}$
    Update $W_u$ and $W_v$
else if an edge $e = (u, v)$ has been deleted then
    Update $W_u$ and $W_v$
end if
Run procedure Fix
```

```
procedure Fix:
while there exists a dirty vertex $v$ do
    if $W_v > c_v$ then
        Increment the level of $v$ by setting $\ell(v) \leftarrow \ell(v) + 1$
        Update $W_v$ and $W_u$ for all affected $v$'s neighboring vertices $u$
    else if $W_v < c_v^*$ and $\ell(v) > 0$ then
        Decrement the level of $v$ by setting $\ell(v) \leftarrow \ell(v) - 1$
        Update $W_v$ and $W_u$ for all affected $v$'s neighboring vertices $u$
    end if
end while
```

### 3.2 Time Complexity

Each update involves two steps, namely the adjustment of weights of the endpoints, and the running of procedure Fix. We now give the time complexity analysis, where the main idea is to prove the following two facts: *(Fact 1)* the amortized time of the adjustment step is $O(\log n)$, and *(Fact 2)* the amortized time of the procedure Fix is zero, irrespective of the number of vertices or edges that are affected during this step. Once the above two facts are proven, the time complexity analysis follows.

We use the standard potential method in our amortized analysis. Imagine that we have a bank account $B$. Initially, the graph is empty, and the bank account $B$ has no potential. For each adjustment step during an edge insertion or deletion, we deposit some potential into the bank account $B$; after that, we use the potential in $B$ to pay for the time of the procedure Fix.

Following the definition of [7], we say a vertex $v \in V$ is *active* if its degree in $G$ is non-zero, and *passive* otherwise. Now, the value of $B$ is set by the following formula:

$$B = \frac{1}{\epsilon} \cdot \left( \sum_{e \in E} \phi(e) + \sum_{v \in V} \psi(v) \right),$$

where $0 < \epsilon < 1$, and $\phi$ and $\psi$ are functions defined as follows:

$$\phi(e) = \left(\frac{\beta}{(\beta-1)} + \epsilon\right)(L - \ell(e)).$$

$$\psi(v) = \begin{cases} \dfrac{\beta^{(\ell(v)+1)}}{\mu(\beta-1)} \cdot \max\{0, \alpha\, c_v^* - W_v\}, & \text{if } v \text{ is } \textit{active}. \\ 0, & \text{otherwise.} \end{cases}$$

The following lemma proves Fact 1.

**Lemma 4** *After the adjustment step, the potential $B$ increases by at most $O(\log n/\epsilon)$.*

***Proof*** We separate the discussion into two cases: edge insertion and edge deletion. Let $t$ be the moment where the update occurs.

- *Edge Insertion* The inserted edge $e$ generates a change of at most $\left(\frac{\beta}{(\beta-1)} + \epsilon\right) L$ in $\phi(e)$. So, the summation $\sum \phi(e)$ increases by at most $O(\log n)$. For each endpoint $v$ of $e$, there are two possible cases for the change in $\psi(v)$:
  - **Case** 1 The vertex $v$ was *passive* at moment $t-1$. By the definition of $\psi(v)$, we had $\psi(v) = 0$ and $\ell(v) = 0$ before the insertion of the edge $e$. Hence, after the insertion of $e$, we have

    $$\begin{aligned} \psi(v) \;&=\; \frac{\beta}{\mu(\beta-1)} \cdot \max\{0, \alpha\, c_v^* - W_v\} \;\le\; \frac{\beta}{\mu(\beta-1)} \cdot \alpha\, c_v^* \\ &\le\; \frac{\beta}{\mu(\beta-1)} \cdot c_v \;<\; \frac{\beta}{\beta-1}. \end{aligned}$$

    Therefore, the summation $\sum \psi(v)$ increases by at most $O(1)$.
  - **Case** 2 The vertex $v$ was *active* at moment $t-1$. In this case, the vertex $v$ remains *active* at moment $t$. Thus, the weight $W_v$ increases, and $\psi(v)$ can only decrease.

  In both cases, the total potential $B$ increases by at most $O(\log n/\epsilon)$ after an edge insertion.

- *Edge Deletion* If an edge $e$ is deleted from $E$, then $\phi(e)$ drops to zero, so that the summation $\sum \phi(e)$ decreases. In contrast, the weight $W_v$ of each endpoint $v$ of $e$ decreases by at most $\mu\beta^{-\ell(v)}$. So, $\psi(v)$ increases by at most

  $$\frac{\beta^{(\ell(v)+1)}}{\mu(\beta-1)} \;\cdot\; \mu\beta^{-\ell(v)} = \frac{\beta}{\beta-1},$$

  which is a constant. Thus, the summation $\sum \psi(v)$ increases by at most $O(1)$. In summary, the total potential $B$ increases by at most $O(1/\epsilon)$ after an edge deletion.

By the above arguments, the lemma follows. □

We now switch our attention to Fact 2. Observe that the procedure FIX performs a series of level up and level down events. For each such event, the level of a specific vertex $v$ will be changed, which will then incur a change in its weight, and changes in the weights of some of the incident edges and their endpoints. The illustration and corresponding lemmas can refer to Fig. 1 first. Let $t_0$ denote the moment before a level up or a level down event, and $t_1$ denote the moment after the weights of the edges and vertices are updated due to this event. Let COUNT denote the number of times an edge in the graph $G$ is updated (for simplicity, we assume that in one edge update, the weight and the assignment of the edge may be updated, and so do the weights of its endpoints, where all these can be done in $O(1)$ time).

For ease of notation, in the following, a superscript $t$ in a variable denotes the variable at moment $t$. For instance, $W_v^{t_0}$ stands for the weight $W_v$ of $v$ at moment $t_0$. Also, we use $\Delta x$ to denote the quantity $x^{t_0} - x^{t_1}$, so that

$$|\Delta\text{COUNT}| = |\text{COUNT}^{t_0} - \text{COUNT}^{t_1}| = \text{COUNT}^{t_1} - \text{COUNT}^{t_0}$$

represents the number of incident edges whose weights are changed between $t_0$ and $t_1$.

Briefly speaking, based on the level scheme and the potential function $B$, we can show:

- For each level up event, each of the affected edges $e$ would have its $\phi(e)$ value dropped, so that an $\epsilon$ fraction can pay for the weight updates of itself and its endpoints, while the remaining fraction can be converted into the increase in $\psi(v)$ value.
- For each level down event, the reverse happens, where the vertex $v$ would have its $\psi(v)$ value dropped, so that an $\epsilon$ fraction can pay for the weight updates of the affected edges and their endpoints, while the remaining fraction can be converted into the increase in $\phi(e)$ values of the affected edges. The $\alpha$ value controls the frequency of the level down events, while trading this off with the approximation guarantee.

Sections 3.2.1 and 3.2.2 present the details of the amortized analysis of these two types of events, respectively. Finally, note that there is no potential input to the bank $B$ after the adjustment step, so that the analysis implies that the procedure FIX must stop (as the potential in the bank is finite).

### 3.2.1 Amortized Time of Level Up

Let $v$ be the vertex that undergoes the level up event, and $i = \ell(v)$ denote its level at moment $t_0$. By our notation, $\Delta B = B^{t_0} - B^{t_1}$ denotes the potential *drop* in the bank $B$ from moment $t_0$ to moment $t_1$. To show that the amortized time of a level up event is at most zero, it is equivalent to show that $\Delta B \geq |\Delta\text{COUNT}|$.

Recall that after a level up event, only the value of $\psi(v)$, the values of $\phi(e)$ and $\psi(u)$ for an edge $(u, v)$ may be affected. In the following, we will examine carefully the changes in these values, and derive the desired bound for $\Delta B$. First, we have the following simple lemma.

**Lemma 5** $|\Delta\textsc{Count}| \le D_v^{t_0}(0, i)$.

***Proof*** When $v$ changes from level $i$ to $i + 1$, only those incident edges with levels $i$ will be affected.

□

The next three lemmas examine, respectively, the changes $\Delta\psi(v)$, $\Delta\phi(e)$, and $\Delta\psi(u)$.

**Lemma 6** $\Delta\psi(v) = 0$.

***Proof*** Since $v$ undergoes a level up event, we have $W_v^{t_0} > c_v > \alpha\, c_v^*$, so that

$$\psi^{t_0}(v) = 0.$$

Next, we look at $\psi^{t_1}(v)$. To begin with, we show a general relationship between $W_v^{t_0}$ and $W_v^{t_1}$, similar to that in the proof of Lemma 3. Let $i$ denote the level $\ell(v)$ of $v$ at moment $t_0$.

- **Case** 1 $D_v^{t_0}(0, i) > k_v$

$$\begin{aligned} W_v^{t_0} &= W_v(i) = k_v \cdot \mu\beta^{-i} + \min\{k_v, D_v^{t_0}(i+1)\} \cdot \mu\beta^{-(i+1)} + \sum_{j > i+1} \min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j} \\ W_v^{t_1} &= W_v(i+1) = k_v \cdot \mu\beta^{-(i+1)} + \sum_{j > i+1} \min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j} \\ &\ge \frac{1}{\beta+1} W_v^{t_0} \end{aligned}$$

- **Case** 2 $D_v^{t_0}(0, i) \le k_v$

$$\begin{aligned} W_v^{t_0} &= W_v(i) = D_v^{t_0}(0, i) \cdot \mu\beta^{-i} + \min\{k_v, D_v^{t_0}(i+1)\} \cdot \mu\beta^{-(i+1)} \\ &\quad + \sum_{j > i+1} \min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j} \\ W_v^{t_1} &= W_v(i+1) = \min\{k_v, D_v^{t_0}(0, i+1)\} \cdot \mu\beta^{-(i+1)} + \sum_{j > i+1} \min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j} \\ &\ge \frac{1}{\beta+1} W_v^{t_0} \end{aligned}$$

Thus, $W_v^{t_1} \ge W_v^{t_0}/(\beta+1) > c_v/(\beta+1) = \alpha\, c_v^*$, which implies

$$\psi^{t_1}(v) = 0.$$

In summary, we have $\Delta\psi(v) = 0 - 0 = 0$ as desired. □

**Lemma 7** *For every edge e incident to v,*

$$\Delta\phi(e) = \begin{cases} \left(\dfrac{\beta}{(\beta-1)} + \epsilon\right), & \textit{if } \ell(e) \in [0, i]. \\ 0, & \textit{otherwise}. \end{cases}$$

***Proof*** As mentioned, only those edges that are at the level in the range $[0, i]$ are affected, so that

$$\begin{aligned}\Delta\phi(u, v) &= \phi^{t_0}(u, v) - \phi^{t_1}(u, v)\\ &= \left(\frac{\beta}{(\beta-1)} + \epsilon\right)(L - i) - \left(\frac{\beta}{(\beta-1)} + \epsilon\right)(L - (i+1))\\ &= \left(\frac{\beta}{(\beta-1)} + \epsilon\right).\end{aligned}$$

□

**Lemma 8** *For every vertex $u \in N_v^{t_0}$, $\Delta\psi(u) \geq -\beta/(\beta-1)$.*

***Proof*** If $\ell(u) \in [i+1, L]$, then $w^{t_0}(u, v) = w^{t_1}(u, v)$ and thus $\Delta w(u, v) = 0$, which implies that $\Delta\psi(u) = 0$. The potential $\psi(u)$ changes only when the level of vertex u is in the range $[0, i]$. Without loss of generality, we assume $\ell(u) = i$ and prove the lemma by considering the relationship between $k_u$, $D_u^{t_0}(0, i)$ and $D_u^{t_0}(i+1)$. For those vertices $u$ with $\ell(u) \in [0, i-1]$, we replace the term $D_u^{t_0}(0, i)$ with $D_u^{t_0}(i)$ while maintaining the same result.

- **Case** $1 D_u^{t_0}(0, i) > k_u$, $D_u^{t_0}(i+1) \geq k_u$

$$W_u^{t_0} = W_u^{t_1} \qquad \Rightarrow \qquad \Delta\psi(u) = 0.$$

- **Case** $2 D_u^{t_0}(0, i) > k_u$, $D_u^{t_0}(i+1) < k_u$

$$\begin{aligned}W_u^{t_1} &= W_u^{t_0} + \mu\beta^{-(i+1)}\\ \Delta\psi(u) &= \frac{\beta^{(\ell(u)+1)}}{\mu(\beta-1)} \cdot \mu\beta^{-(i+1)} = \frac{1}{\beta-1} \cdot \beta^{\ell(u)-i} > 0.\end{aligned}$$

- **Case** $3 D_u^{t_0}(0, i) \leq k_u$, $D_u^{t_0}(i+1) < k_u$

$$\begin{aligned}W_u^{t_1} &= W_u^{t_0} - \mu(\beta^{-i} - \beta^{-(i+1)})\\ \Delta\psi(u) &= -\frac{\beta^{(\ell(u)+1)}}{\mu(\beta-1)} \cdot \mu(\beta^{-i} - \beta^{-(i+1)}) = -\frac{\beta^{(\ell(u)+1)}}{\beta^{i+1}} \geq -1.\end{aligned}$$

- **Case** $4 D_u^{t_0}(0, i) \leq k_u$, $D_u^{t_0}(i+1) \geq k_u$

$$\begin{aligned}W_u^{t_1} &= W_u^{t_0} - \mu\beta^{-i}\\ \Delta\psi(u) &= -\frac{\beta^{(\ell(u)+1)}}{\mu(\beta-1)} \cdot \mu\beta^{-i} = -\frac{\beta}{\beta-1}(\beta^{\ell(u)-i}) \geq -\frac{\beta}{\beta-1}.\end{aligned}$$

□

Based on the above lemmas, we derive the following and finish the proof for the case of level up.

$$\begin{aligned}\Delta B &= \frac{1}{\epsilon}\cdot\left(\Delta\psi(v)+\sum_{e\in E}\Delta\phi(e)+\sum_{u\in N_v^{t_0}}\Delta\psi(u)\right)\\ &\geq \frac{1}{\epsilon}\cdot\left(0+\left(\frac{\beta}{(\beta-1)}+\epsilon\right)D_v^{t_0}(0,i)-\frac{\beta}{\beta-1}D_v^{t_0}(0,i)\right)\\ &= D_v^{t_0}(0,i) \;\geq\; |\Delta\textsc{Count}|.\end{aligned}$$

### 3.2.2 Amortized Time of Level Down

We now show that the amortized time of level down for a vertex $v$ is at most zero. Similar to the case of level up, we examine $\Delta\psi(v)$, $\Delta\phi(e)$, and $\Delta\psi(u)$, and show that $\Delta B \geq |\Delta\textsc{Count}|$.

Before starting the proof of the level down case, recall that we have mentioned the parameter $h$ at the end of the introduction, where $h$ is the largest number of selected copies of all the vertices. That is, $h = \max_v\{\lceil|\delta^{t_0}(v)|/k_v\rceil\}$. Also, we let $h' = \max_v\{\lceil D_v^{t_0}(0,\ell(v))/k_v\rceil\}$, where $h' \geq h$, and set $\xi \geq 0$ such that $h' = h + \xi$.

**Lemma 9** $|\Delta\textsc{Count}| \leq D_v^{t_0}(0,i) < h'\cdot\frac{\beta^i c_v^*}{\mu}$.

***Proof*** When the vertex $v$ moves from level $i$ to $i-1$, only those edges whose levels are at most $i$ are affected. This shows the first part of the inequality. Also, because $v$ undergoes level down, we have $W_v^{t_0} < c_v^*$. Then, for the latter inequality, we partition the proof into two cases:

- **Case** 1 $\lceil|\delta^{t_0}(v)|/k_v\rceil = 1$

$$\begin{aligned}W_v^{t_0} &= D_v^{t_0}(0,i)\cdot\mu\beta^{-i}+\sum_{j>i}\min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j}\\ \Rightarrow\; c_v^* &> D_v^{t_0}(0,i)\cdot\mu\beta^{-i}\\ \Rightarrow\; D_v^{t_0}(0,i) &< \frac{\beta^i c_v^*}{\mu}.\end{aligned}$$

- **Case** 2 $\lceil|\delta^{t_0}(v)|/k_v\rceil > 1$

$$\begin{aligned}W_v^{t_0} &= k_v\cdot\mu\beta^{-i}+\sum_{j>i}\min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j}\\ \Rightarrow\; c_v^* &> k_v\cdot\mu\beta^{-i}\end{aligned}$$

$$\Rightarrow h'c_v^* > \frac{D_v^{t_0}(0,i)}{k_v} \cdot k_v \cdot \mu\beta^{-i}$$
$$\Rightarrow D_v^{t_0}(0,i) < h' \cdot \frac{\beta^i c_v^*}{\mu}.$$

□

Now, we are ready to examine $\Delta\psi(v)$, $\Delta\phi(e)$, and $\Delta\psi(u)$, through the following lemmas.

**Lemma 10** *For every vertex $u \in N_v^{t_0}$, $\Delta\psi(u) \geq -1/(\beta - 1)$.*

***Proof*** If $\ell(u) \in [i, L]$, then $w^{t_0}(u, v) = w^{t_1}(u, v)$ and $\Delta w(u, v) = 0$, which implies $\Delta\psi(u) = 0$. The changes of potentials only occur at the vertex whose level is in the range $[0, i-1]$. Without loss of generality, we assume $\ell(u) = i - 1$ and we consider the relationship between $k_u$, $D_u^{t_0}(0, i-1)$ and $D_u^{t_0}(i)$. For those vertices $u$ and $\ell(u) \in [0, i-2]$, we replace the term $D_u^{t_0}(0, i-1)$ with $D_u^{t_0}(i-1)$ while maintaining the same result.

- **Case** 1 $D_u^{t_0}(0, i-1) \geq k_u$, $D_u^{t_0}(i) > k_u$

$$W_u^{t_0} = W_u^{t_1} \Rightarrow \Delta\psi(u) = 0.$$

- **Case** 2 $D_u^{t_0}(0, i-1) \geq k_u$, $D_u^{t_0}(i) \leq k_u$

$$\begin{aligned} W_u^{t_1} &= W_u^{t_0} - \mu\beta^{-i} \\ \Delta\psi(u) &= -\frac{\beta^{(\ell(u)+1)}}{\mu(\beta-1)} \cdot \mu\beta^{-i} \\ &= -\frac{\beta}{\beta-1} \cdot \beta^{\ell(u)-i} \qquad (\because \ell(u) \leq i-1) \\ &\geq -\frac{1}{\beta-1}. \end{aligned}$$

- **Case** 3$D_u^{t_0}(0, i-1) < k_u$, $D_u^{t_0}(i) > k_u$

$$W_u^{t_1} = W_u^{t_0} + \mu\beta^{-(i-1)}$$
$$\Rightarrow \psi^{t_0}(u) > \psi^{t_1}(u) \Rightarrow \Delta\psi(u) > 0.$$

- **Case** 4$D_u^{t_0}(0, i-1) < k_u$, $D_u^{t_0}(i) \leq k_u$

$$W_u^{t_1} = W_u^{t_0} + \mu(\beta^{-(i-1)} - \beta^{-i})$$
$$\Rightarrow \psi^{t_0}(u) > \psi^{t_1}(u) \Rightarrow \Delta\psi(u) > 0.$$

□

Next, we partition $N_v^{t_0}$ into three subsets: $X$, $Y_1$ and $Y_2$, i.e. $N_v^{t_0} = X \cup Y_1 \cup Y_2$, where

$$\begin{aligned} X &= \{u \mid u \in N_v^{t_0}(0, i-1)\}, \\ Y_1 &= \{u \mid u \in N_v^{t_0}(i)\}, \\ Y_2 &= \{u \mid u \in N_v^{t_0}(i+1, L)\}. \end{aligned}$$

**Lemma 11** *For every edge $(u, v)$ incidents to a vertex $v$,*

$$\Delta\phi(u, v) = \begin{cases} -\left(\frac{\beta}{(\beta-1)} + \epsilon\right), & \textit{if } u \in X \\ 0, & \textit{if } u \in Y_1 \cup Y_2. \end{cases}$$

***Proof*** Fix any vertex $u \in N_v^{t_0}$. We consider the following two possible scenarios.

- **Case** $1 u \in Y_1 \cup Y_2$ When the level of the vertex $v$ decreases from $i$ to $i-1$, $\ell^{t_0}(u, v) = \ell^{t_1}(u, v)$ and thus $\phi^{t_0}(u, v) = \phi^{t_1}(u, v)$, which implies $\Delta\phi(u, v) = 0$.
- **Case** $2\ u \in X$
  When the level of the vertex $v$ decreases from $i$ to $i-1$, we have $\ell^{t_0}(u, v) = i$ and $\ell^{t_1}(u, v) = (i-1)$. The following result is thus derived:

$$\begin{aligned} \Delta\phi(u, v) &= \left(\frac{\beta}{(\beta-1)} + \epsilon\right)(L-i) - \left(\frac{\beta}{(\beta-1)} + \epsilon\right)(L-i+1) \\ &= -\left(\frac{\beta}{(\beta-1)} + \epsilon\right). \end{aligned}$$

□

Next, let $W_v^{t_0} = x + y_1 + y_2$, where $x$, $y_1$ and $y_2$ on the right-hand-side correspond to the weights generated by the subsets $X$, $Y_1$, $Y_2$, respectively. So, we get the following lemmas:

**Lemma 12** $\sum_{u \in N_v^{t_0}} \Delta\phi(u, v) \leq -\left(\frac{\beta}{(\beta-1)} + \epsilon\right)(hx\beta^i/\mu)$.

***Proof*** We consider $|X|$ in the following two cases:

- **Case** $1 |X| \leq k_v$

$$x = |X| \cdot \mu\beta^{-i} \Rightarrow |X| = \frac{x\beta^i}{\mu} \leq hx\beta^i/\mu \quad (\because \text{h} \geq 1)$$

- **Case** $2 |X| > k_v$. Here, we may assume, without loss of generality, that $y_1 = 0$. Then, we have

$$\begin{aligned} & x = k_v \cdot \mu\beta^{-i} \\ \Rightarrow\ & |X| = \frac{|X|}{k_v} \cdot \frac{x\beta^i}{\mu} \leq \left\lceil \frac{\delta^{t_0}(v)}{k_v} \right\rceil \cdot \frac{x\beta^i}{\mu} \leq h \cdot \frac{x\beta^i}{\mu}. \end{aligned}$$

Finally, since

$$\sum_{u \in N_v^{t_0}} \Delta\phi(u,v) = |X| \cdot -\left(\frac{\beta}{(\beta-1)} + \epsilon\right),$$

the lemma thus follows. □

**Lemma 13** $\Delta\psi(v) = (\alpha c_v^* - x - y_1 - y_2) \cdot \frac{\beta^{i+1}}{\mu(\beta-1)} - \max\{0, \alpha c_v^* - \beta x - y_1 - y_2\} \cdot \frac{\beta^i}{\mu(\beta-1)}$.

***Proof*** We have $W_v^{t_0} = x + y_1 + y_2 < c_v^*$, and we have to consider the following relationship between $x + y_1$ and $k_v \cdot \mu\beta^{-i}$. With the above relationship, we compute $W_v^{t_1}$ by the following:

- **Case** 1 $|X| < k_v$ and $|X + Y_1| \le k_v$:

$$\begin{aligned} W_v^{t_0} &= W_v(i) = D_v^{t_0}(0,i)\mu\beta^{-(i)} + \sum_{j>i} \min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j}; \\ W_v^{t_1} &= W_v(i-1) = D_v^{t_0}(0,i-1)\mu\beta^{-(i-1)} + D_v^{t_0}(i)\mu\beta^{-(i)} + \sum_{j>i+1} \min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j} \\ &= \beta x + y_1 + y_2; \end{aligned}$$

- **Case** 2 $|X + Y_1| > k_v$:

$$\begin{aligned} W_v^{t_0} &= W_v(i) = k_v\mu\beta^{-(i)} + \sum_{j>i} \min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j}; \\ W_v^{t_1} &= W_v(i-1) \\ &= \min\{k_v, D_v^{t_0}(0,i-1)\}\mu\beta^{-(i-1)} + \min\{k_v, D_v^{t_0}(i)\}\mu\beta^{-(i)} + \sum_{j>i} \min\{k_v, D_v^{t_0}(j)\}\mu\beta^{-j} \\ &\le (\beta+1)x + y_1 + y_2; \end{aligned}$$

By the above cases, we have a weight change of at least $\beta x + y_1 + y_2$ in $W_v$. The desired bound on $\Delta\psi(v)$ can thus be obtained by direct substitution. □

Finally, depending upon the value of $\alpha c_v^* - \beta x - y_1 - y_2$, we consider two possible scenarios, where we show that in each case, $\Delta B \ge h' \cdot \beta^i c_v^*/\mu$. This in turn implies $\Delta B \ge |\Delta\text{COUNT}|$ as desired.

- **Case** $1\alpha c_v^* \le \beta x + y_1 + y_2$

$$
\begin{aligned}
\epsilon \cdot \Delta B &= \left( \sum_{u \in N_v^{t_0}} \Delta\psi(u) + \sum_{e \in E} \Delta\phi(e) + \Delta\psi(v) \right) \\
&\ge -\frac{1}{\beta - 1} \cdot \frac{hx\beta^i}{\mu} - \left( \frac{\beta}{(\beta-1)} + \epsilon \right) \cdot \frac{hx\beta^i}{\mu} + (\alpha c_v^* - x - y_1 - y_2) \cdot \frac{\beta^{i+1}}{\mu(\beta - 1)} \\
&\ge \frac{\beta^i}{\mu} \left( -\frac{1}{\beta - 1} h c_v^* - \left( \frac{\beta}{(\beta-1)} + \epsilon \right) h c_v^* + \frac{(\alpha - 1)\beta}{\beta - 1} c_v^* \right) \quad (\because c_v^* \ge x + y_1 + y_2) \\
&= \frac{\beta^i c_v^*}{\mu(\beta - 1)} ((\alpha - 1)\beta - h - (\beta + (\beta - 1)\epsilon) h) \\
&= \frac{\beta^i c_v^*}{\mu} \left( \frac{(\alpha - 1)\beta}{(\beta - 1)} - h \left( \frac{\beta + 1}{\beta - 1} + \epsilon \right) \right) \quad \text{if let } \alpha = \frac{\beta - 1}{\beta} \left( h \left( \frac{\beta+1}{\beta-1} + 2\epsilon \right) + \xi\epsilon \right) + 1 \\
&\ge \epsilon h' \cdot \frac{\beta^i c_v^*}{\mu}.
\end{aligned}
$$

- **Case** 2 $\alpha c_v^* > \beta x + y_1 + y_2$

$$
\begin{aligned}
\epsilon \cdot \Delta B &= \left( \sum_{u \in N_v^{t_0}} \Delta\psi(u) + \sum_{e \in E} \Delta\phi(e) + \Delta\psi(v) \right) \\
&\ge -\frac{1}{\beta - 1} \cdot \frac{hx\beta^i}{\mu} - \left( \frac{\beta}{(\beta-1)} + \epsilon \right) \cdot \frac{hx\beta^i}{\mu} + (\alpha c_v^* - x - y_1 - y_2) \cdot \frac{\beta^{i+1}}{\mu(\beta - 1)} \\
&\quad - (\alpha c_v^* - \beta x - y_1 - y_2) \cdot \frac{\beta^i}{\mu(\beta - 1)} \\
&= \frac{\beta^i}{\mu(\beta - 1)} \cdot (-xh - (\beta + (\beta - 1)\epsilon) xh + \alpha(\beta - 1) c_v^* - (\beta - 1)(y_1 + y_2)) \\
&= \frac{\beta^i}{\mu(\beta - 1)} \cdot (\alpha(\beta - 1) c_v^* - (\beta + 1 + (\beta - 1)\epsilon) xh - (\beta - 1)(y_1 + y_2)) \\
&\ge \frac{\beta^i c_v^*}{\mu} \cdot \left( \alpha - h \left( \frac{\beta + 1}{\beta - 1} + \epsilon \right) \right) \quad \text{if let } \alpha = \frac{\beta - 1}{\beta} \left( h \left( \frac{\beta+1}{\beta-1} + 2\epsilon \right) + \xi\epsilon \right) + 1 \\
&\ge \epsilon h' \cdot \frac{\beta^i c_v^*}{\mu}.
\end{aligned}
$$

Thus, the level scheme remains $\alpha(\beta + 1)$-tight after a level down event. However, the value of $h$ is bounded by $n$, and $h$ appears inside $\alpha$, so that the approximation ratio of the scheme may become $n$ in the worst-case. Fortunately, with the help of the following lemma, we can choose $\alpha$ carefully, which in turn improves the approximation ratio from $n$ to $O(1)$.

**Lemma 14** *Suppose we set* $\alpha \ge \beta/(\beta - 1)$. *By the time a level down event occurs at* $v$ *at moment* $t_0$, *exactly one copy of* $v$ *is selected. That is,* $\lceil |\delta^{t_0}(v)|/k_v \rceil = 1$.

***Proof*** Assume to the contrary that $v$ could decrease its level even if more than one copy of $v$ is selected. Since $v$ levels down, its weight $W_v$ must have decreased; this can happen only in one of the following cases:

- **Case** 1 An incident edge whose level is in the range $[0, \ell(v)]$ is deleted. In this case, since more than one copy of $v$ is selected, $W_v$ is unchanged. Thus, this case cannot happen.
- **Case** 2 An incident edge whose level is in the range $[\ell(v)+1, L]$ is deleted. In this case, the weight $W_v^{t_0}$ at moment $t_0$ is less than $c_v^*$. On the other hand, at the moment $t'$ when $v$ attains the current level $\ell(v)$ (from level $\ell(v)-1$), its weight $W_v^{t'}$ was at least $c_v$ before it leveled up, and became at least $c_v/(\beta+1)$ after it leveled up.

  (The reason is from the proof of Lemma 3: the weight change between consecutive levels is at most a factor of $\beta+1$.) This implies that:

  $$c_v^* \;>\; W_v^{t_0} \;\geq\; k_v \mu \beta^{-\ell(v)}$$

  The above relation is true because $v$ has to decrease its level at moment $t_0$, which implies the first inequality.

  Next, the second inequality holds because more than one copy of $v$ is selected. Also, we have the following relation:

  $$(\beta/(\beta-1))k_v\mu\beta^{-\ell(v)} \;\geq W_v^{t'} \;\geq\; c_v/(\beta+1)$$

  To see why the above is true, notice that the first inequality holds since the leftmost term is the maximum possible value of $W_v$, while the second inequality holds naturally with the non-zero level of $v$.

  Combining the above two inequalities, we would have

  $$\frac{c_v}{\alpha(\beta+1)} \;=\; c_v^* \;>\; k_v\mu\beta^{-\ell(v)} \;\geq\; \frac{c_v(\beta-1)}{\beta(\beta+1)},$$

  so that $\alpha < \beta/(\beta-1)$. Which leads to a contradiction

Thus, the lemma follows. □

The above lemma states that if we choose $\alpha \geq \beta/(\beta-1)$, then level down of $v$ occurs only when $\lceil |\delta^{t_0}(v)|/k_v \rceil$ is one. Then, Case 2 inside the proof of Lemma 9 will not occur, so that we can strengthen Lemma 9 to get $|\Delta\textsc{Count}| \leq D_v^{t_0}(0,i) < \beta^i c_v^*/\mu$. Similarly, the proof of Lemma 12 can be revised, so that we can strengthen Lemma 12 by replacing $h$ with one. On the other hand, we need $\alpha \geq (2\beta+1)/\beta + 2\epsilon$ to satisfy the amortized time analysis. Consequently, we set $\alpha = (2\beta+1)/\beta + 2\epsilon$, and we can achieve the desired bound $\Delta B \geq \beta^i c_v^*/\mu \geq |\Delta\textsc{Count}|$. The proof for the level down case is complete.

### 3.3 Summary and Extensions

With the appropriate setting of $\alpha = (2\beta+1)/\beta + 2\epsilon$, where $0 < \epsilon < 1$, we get an $\alpha(\beta+1)$-tight level scheme. Then, by setting $\beta = 2.43$, Theorem 2 is proven so that we get an approximate solution of ratio close to 36 with $O((\log n)/\epsilon)$ amortized update time. Finally, we consider two natural extensions of the capacitated vertex cover problem, and show how to adapt the proposed level scheme to handle these extensions

*Capacitated Set Cover* Here, we consider the capacitated set cover problem which is equivalent to the capacitated vertex cover problem in hyper-graphs. A hyper-graph $G = (V, E)$ has $|V| = n$ vertices and $|E| = m$ hyper-edges, where each hyper-edge is incident to a set of vertices. Suppose that each hyper-edge is incident to at most $f$ vertices. Our target is to find a subset of vertices, each with a certain number of copies, so that every edge in $E$ is covered, while the total cost of the selected vertices (each of which is weighted by the corresponding number of copies) is minimized. Here, we treat the hyper-graph vertex cover problem as if the original vertex cover problem, and use the same level scheme and the definition of the weight of a vertex $W_v$. That is, the weight $W_v$ of a vertex $v$ is defined as follows:

- **Case** 1 $D_v(0, \ell(v)) > k_v$:

$$W_v = k_v \mu \beta^{-\ell(v)} + \sum_{i > \ell(v)} \min\{k_v, D_v(i)\} \mu \beta^{-i}$$

- **Case** 2 $D_v(0, \ell(v)) \leq k_v$:

$$W_v = D_v(0, \ell(v)) \mu \beta^{-\ell(v)} + \sum_{i > \ell(v)} \min\{k_v, D_v(i)\} \mu \beta^{-i}$$

We also use the same conditions for level up and level down. However, we still need to do some adjustments for this problem. First, we re-design the number of levels, $L$, to be $\lceil \log_\beta (m\mu\alpha/c_{\min}) \rceil$. Next, we adjust the flexible range by multiplying it by $f$ so that $W_v \in (c_v/f\varepsilon, c_v]$. In Lemma 2, we have proved that if there are more than $k_v$ edges assigned to a vertex $v$, then every edge is accounted for at most $2(\beta/(\beta-1))\pi_e$. But here, a hyper-edge $e$ may be incident to at most $f$ vertices so that the total time for a hyper-edge is bounded by at most $(2(\beta/(\beta-1)) + (f-1))\pi_e$ instead.

By the arguments of Sect. 2, we observe that the approximation ratio comes from (1) the flexible range of *weight* function $W_v$ and (2) the total number of times an edge is used. Since both of these terms are increased by a factor of $f$ in the dynamic set cover problem; thus, the approximation ratio here becomes $O(f^2)$.

When we consider the updated time in the dynamic setting, we modify our potential function as follows:

$$\phi(e) = \left(\frac{\beta}{(\beta-1)} + \epsilon\right)(L - \ell(e)).$$

$$\psi(v) = \begin{cases} \dfrac{\beta^{(\ell(v)+1)}}{f\mu(\beta-1)} \cdot \max\{0, f\alpha\, c_v^* - W_v\}, & \text{if } v \text{ is } \textit{active}. \\ 0, & \text{otherwise.} \end{cases}$$

From Sect. 3, we know that the update time in the dynamic vertex cover problem is related to the number of levels in the level scheme. Here, this number will be adjusted to $O(\log(m+n))$. Furthermore, a hyper-edge is incident to at most $f$ vertices (instead of at most two vertices in the vertex cover problem). Thus, it will affect at most $f$ vertices when there is an edge insertion or deletion. Combining these, we can readily show that our scheme achieves $O(f \log(m+n))$ amortized update time.

*Capacitated Vertex Cover with Non-uniform Unsplittable Demand* In this part, we consider a more general model in which each edge has an unsplittable demand. That is, the demand of each edge must be covered by exactly one of its endpoints. We first show that, in a static setting, with some modification, our approach in Sect. 2 already gives an $O(1)$-approximate solution. First, when we consider the general case, we have to revise the capacity constraint in the primal problem to $k_v x_v - \sum_{e \in N_v} y_{ev} d_e \geq 0$, and we also have to change the vertex constraint in the dual problem to $q_v d_e + l_{ev} \geq \pi_e$.

To cope with these changes, we will revise the number of levels of our level scheme to be $L = \lceil \log_\beta (k_{\max} \mu\alpha / c_{\min}) \rceil$, where $k_{\max}$ denotes the maximum capacity of a vertex. Moreover, we adjust our definition of the weight $W_v$ of a vertex as follows:

- **Case** 1 $\sum_{e \,|\, e \sim v, \ell(e)=\ell(v)} d_e > k_v$:

$$W_v = k_v \mu \beta^{-\ell(v)} + \sum_{j \,|\, \ell(e)=j>\ell(v)} \min\{k_v, \sum_e d_e\} \mu \beta^{-j}$$

- **Case** 2 $\sum_{e \,|\, e \sim v, \ell(e)=\ell(v)} d_e \leq k_v$:

$$W_v = \sum_{e \,|\, e \sim v, \ell(e)=\ell(v)} d_e \mu \beta^{-\ell(v)} + \sum_{j \,|\, \ell(e)=j>\ell(v)} \min\{k_v, \sum_e d_e\} \mu \beta^{-j}$$

where $e \sim v$ denotes $e$ is an edge incident to $v$.

Due to the change of the mathematical model in both primal and dual problems, we need a slightly different strategy from that in Sect. 2. We use the total demand of the unassigned edges to replace the number of unassigned edges to determine the value of $q_v$ and $l_{ev}$. In particular:

If $\lceil \sum_{e \in \delta(v)} d_e / k_v \rceil > 1$: $q_v = \mu \beta^{-\ell(v)}$, and $l_{ev} = 0$;
If $\lceil \sum_{e \in \delta(v)} d_e / k_v \rceil \leq 1$: $q_v = \mu \sum_{i \,|\, \sum_{\ell(e)=i} d_e \geq k_v} \beta^{-i}$, $l_{ev} = 0$ if $\sum_{\ell(e)=i} d_e \geq k_v$, and $l_{ev} = d_e \cdot \mu \beta^{-\ell(e)}$ otherwise.
For every edge $e$: $\pi_e = d_e \cdot \mu \beta^{-\ell(e)}$.

Then, we use the same technique as that in Sect. 2, and it is easy to verify that the above choices of $q_v$, $l_{ev}$, and $\pi_e$ give a feasible solution to the dual problem. Again, for the total cost of our solution, we separate the analysis into two parts, based on the multiplicity of the vertex $v$:

- **Case** 1 $\lceil \sum_{e \in \delta(v)} d_e / k_v \rceil > 1$: In this case, the external component of $W_v$ is at most $1/(\beta - 1)$ of the internal component, so that $W_v \leq (\beta/(\beta - 1)) k_v q_v$. Then, the cost of all copies of $v$ is:

$$\begin{aligned}\left\lceil \sum_{e\in\delta(v)} d_e/k_v \right\rceil \cdot c_v &\le \left\lceil \sum_{e\in\delta(v)} d_e/k_v \right\rceil \cdot \varepsilon \cdot W_v \\ &\le 2\cdot \frac{\sum_{e\in\delta(v)} d_e}{k_v} \cdot \varepsilon \cdot (\beta/(\beta-1))k_v q_v \\ &= 2(\beta/(\beta-1))\varepsilon \cdot \sum_{e\in\delta(v)} d_e q_v \;=\; 2(\beta/(\beta-1))\varepsilon \cdot \sum_{e\in\delta(v)} \pi_e.\end{aligned}$$

- **Case** 2 $\lceil \sum_{e\in\delta(v)} d_e/k_v \rceil = 1$: In this case, we pick one copy of vertex $v$, whose cost is:

$$\begin{aligned}c_v &\le \varepsilon \cdot W_v \\ &\le \varepsilon \cdot \sum_{e\sim v} \pi_e \;=\; \varepsilon \cdot \left( \sum_{e\in\delta(v)} \pi_e + \sum_{e\notin\delta(v),\, e\sim v} \pi_e \right),\end{aligned}$$

As compared to the uniform demand case, every edge multiplies its demand. Yet, the selected copies also multiply the same constant. Thus, with an analogous analysis, the approximation ratio of the revised algorithm in this section (for non-uniform demand) is the same for the uniform demand case.

Unfortunately, when we consider the dynamic operations, an edge insertion or deletion may cause a vertex to adjust its level severely because the edge weight, in this case, connects to the edge's demand. It is open whether we can maintain a constant approximation ratio with polylogarithmic update time for this general problem where edges have non-uniform unsplittable demands.

However, we still present two simple approaches for this problem by combining other techniques with the initially proposed level scheme in Sect. 2.

The first approach is to partition all of the edges into $\log_2(d_{\max})$ clusters according to its demand (where the $i$th cluster contains edges with demand in the range $[2^{i-1}, 2^i)$), and maintain each cluster by its own data structure. In every cluster, we set value of $\alpha = 2((2\beta+1)/\beta + 2\epsilon)$. Whenever there is an edge insertion or edge deletion, we run the proposed algorithm in the corresponding cluster. That is, only the data structure of one cluster is updated per each edge update event. For the output, we select the vertices, and their corresponding number of copies, in each of the clusters to cover all the edges in that cluster. After these changes, we obtain an $O(\log d_{\max})$ approximation ratio solution with $O(L/\epsilon) = O(\log k_{\max}/\epsilon)$ update time, where $d_{\max} = \max_e\{d_e\}$ and $k_{\max} = \max_v\{k_v\}$.

The second approach works for integral demands. We view an edge $e$ with demand $d$ as $d$ edges $e_1, e_2, \ldots, e_d$ with uniform demand between the same endpoints. Then, we execute the proposed level scheme. The only problem is that those edges $e_1, e_2, \ldots, e_d$ corresponding to the original edge $e$ may be assigned to the different endpoints. We simply assign *all* edges to the endpoint that is covering the majority of these edges, based on the solution in the proposed level scheme. After that, the total time is increased by at most a factor of 2, so that we obtain an O(1)-approximate solution with the $O(d_{\max} L/\epsilon) = O(d_{\max} \log k_{\max}/\epsilon)$ amortized update time.

## 4 Concluding Remarks

We have extended dynamic vertex cover to the more general WMCVC problem, and developed a constant-factor dynamic approximation algorithm with $O(\log n/\epsilon)$ amortized update time, where $n$ is the number of the vertices. Note that, in Gupta et al.'s very recent paper [12], their greedy algorithm with minor adaptions is also able to work for the soft dynamic capacitated vertex cover problem. However, it only gives a *logarithmic*-factor approximation algorithm with $O(\log n)$ amortized update time. Moreover, our proposed algorithm can also be extended to solve the (soft) capacitated set cover problem, and the soft capacitated vertex cover problem with non-uniform unsplittable edge demand.

We conclude this paper with some open problems. First, recall that in the *static* model, the soft capacitated vertex cover problem [11] can be approximated within a factor of two and three for the uniform and non-uniform edge demand cases, respectively. Here, we have shown that it is possible to design a dynamic scheme with $O(1)$ approximation ratio with polylogartihmic update time for the uniform edge demand case. Thus, designing an $O(1)$-approximation ratio algorithm with $O(\log k_{\max})$, or polylogarithmic, update time for the non-uniform edge demand case seems promising.

Moreover, it would also be of significant interest to explore whether it is possible to derive a constant approximation ratio for the WMCVC problem under constant update time. Also, in recent years, more studies on the worst-case update time for dynamic algorithms have been conducted. It would be worthwhile to examine update time in the worst-case analysis.

**Acknowledgements** The extended abstract of this paper has been published in Proceedings of the 21st International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'2018). We also thank the support from National Center for High-performance Computing (NCHC) and the Brain Research Center under the Higher Education Sprout Project.

## References

1. Andersson, A., Thorup, M.: Dynamic ordered sets with exponential search trees. J. ACM **54**(3), 13 (2007)
2. Archetti, C., Feillet, D., Mor, A., Speranza, M.G.: Dynamic traveling salesman problem with stochastic release dates. Eur. J. Oper. Res. **280**(3), 832–844 (2020)
3. Baswana, S., Gupta, M., Sen, S.: Fully dynamic maximal matching in $O(\log n)$ update time. SIAM J. Comput. **44**(1), 88–113 (2015)
4. Bhattacharya, S., Chakrabarty, D., Henzinger, M.: Fully dynamic approximate maximum matching and minimum vertex cover in $O(\log^3 n)$ worst case update time. In: Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA), Barcelona, Spain, pp. 470–489 (2017)
5. Bhattacharya, S., Chakrabarty, D., Henzinger, M.: Deterministic fully dynamic approximate vertex cover and fractional matching in $O(1)$ amortized update time. In: Proceedings of the 19th Conference on Integer Programming and Combinatorial Optimization (IPCO), Waterloo, Canada, pp. 86–98 (2017)

6. Bhattacharya, S., Henzinger, M., Italiano, G.F.: Deterministic fully dynamic data structures for vertex cover and matching. In: Proceedings of the 26th ACM-SIAM Symposium on Discrete Algorithms (SODA), Philadelphia, USA, pp. 785–804 (2015)
7. Bhattacharya, S., Henzinger, M., Italiano, G.F.: Design of dynamic algorithms via primal-dual method. In: Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP), Heidelberg, Germany, vol. 2015, pp. 206–218 (2015)
8. Buriol, L.S., Resende, M.G.C., Thorup, M.: Speeding up dynamic shortest-path algorithms. Inf. J. Comput. **20**(2), 191–204 (2008)
9. Chuzhoy, J., Naor, J.: Covering problems with hard capacities. SIAM J. Comput. **36**(2), 498–515 (2006)
10. Demetrescu, C., Italiano, G.F.: A new approach to dynamic all pairs shortest paths. J. ACM **51**(6), 968–992 (2004)
11. Guha, S., Hassin, R., Khuller, S., Or, E.: Capacitated vertex covering. J. Algorithms **48**(1), 257–270 (2003)
12. Gupta, A., Krishnaswamy, R., Kumar, A., Panigrahi, D.: Online and dynamic algorithms for set cover. In: Proceedings of the 49th ACM Symposium on Theory of Computing (STOC), Montreal, Canada, pp. 537–550 (2017)
13. Holm, M.T.J., de Lichtenberg, K.: Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. J. ACM **48**(4), 723–760 (2001)
14. Ivkovic, Z., Lloyd, E.L.: Fully dynamic maintenance of vertex cover. In: Proceedings of the 19th International Workshop on Graph-theoretic Concepts in Computer Science (WG), London, UK, pp. 99–111 (1994)
15. Kejlberg-Rasmussen, C., Kopelowitz, T., Pettie, S., Thorup, M.: Faster worst case deterministic dynamic connectivity. In: Proceedings of the 24th Annual European Symposium on Algorithms (ESA), Aarhus, Denmark, vol. 57, pp. 1–15 (2016)
16. Neiman, O., Solomon, S.: Simple deterministic algorithms for fully dynamic maximal matching. ACM Trans. Algorithms **12**(1), 7 (2016)
17. Onak, K., Rubinfeld, R.: Maintaining a large matching and a small vertex cover. In: Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC), Cambridge, USA, pp. 457–464 (2010)
18. Peleg, D., Solomon, S.: Dynamic ($1+\epsilon$)-approximate matchings: a density-sensitive approach. In: Proceedings of the 27th ACM-SIAM Symposium on Discrete Algorithms (SODA), Virginia, USA, pp. 712–729 (2016)
19. Silva, A., Aloise, D., Coelho, L.C., Rocha, C.: Heuristics for the dynamic facility location problem with modular capacities. Eur. J. Oper. Res. **290**, 435–452 (2020)
20. Solomon, S.: Fully dynamic maximal matching in constant update time. In: Proceedings of the 57th Symposium on Foundations of Computer Science (FOCS), New Jersey, USA, pp. 325–334 (2016)
21. Thorup, M.: Worst-case update times for fully-dynamic all-pairs shortest paths. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC), Baltimore, MD, USA, pp. 112–119 (2005)
22. Yu, G., Yang, Y.: Dynamic routing with real-time traffic information. Oper. Res. **19**, 1–26 (2017)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.