Runtime Performances of Randomized Search Heuristics for the Dynamic Weighted Vertex Cover Problem

Feng Shi1 · Frank Neumann2 · Jianxin Wang1

Received: 14 December 2018 / Accepted: 9 December 2019 / Published online: 20 January 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
Randomized search heuristics such as evolutionary algorithms are frequently applied to dynamic combinatorial optimization problems. Within this paper, we present a dynamic model of the classic weighted vertex cover problem and analyze the runtime performances of the well-studied algorithms randomized local search and \((1 + 1)\) EA adapted to it, to contribute to the theoretical understanding of evolutionary computing for problems with dynamic changes. In our investigations, we use an edge-based representation based on the dual form of the Linear Programming formulation for the problem and study the expected runtime that the adapted algorithms require to maintain a 2-approximate solution when the given weighted graph is modified by an edge-editing or weight-editing operation. Considering the weights on the vertices may be exponentially large with respect to the size of the graph, the step size adaption strategy is incorporated, with or without the 1/5-th rule that is employed to control the increasing/decreasing rate of the step size. Our results show that three of the four algorithms presented in the paper can recompute 2-approximate solutions for the studied dynamic changes in polynomial expected runtime, but the \((1 + 1)\) EA with 1/5-th rule requires pseudo-polynomial expected runtime.

Keywords Runtime analysis · Dynamic weighted vertex cover problem · Graph-editing operation · Evolutionary algorithm · Randomized local search
1 Introduction

Over the past decades, randomized search heuristics such as evolutionary algorithms and ant colony optimization have been applied successfully in various areas, including engineering and economics. To gain a deep insight into the behaviors of evolutionary algorithms, many theoretical techniques for analyzing their expected runtime were presented [1, 12, 18]. And using these techniques, evolutionary algorithms designed for some classic combinatorial optimization problems have been studied. In particular, the Vertex Cover problem plays a crucial role in the area [7, 9, 13, 17, 23].

Consider an instance $I$ of a given combinatorial optimization problem, and a solution $S$ to $I$ satisfying a specific quality guarantee (optimal or approximated). If an operation on $I$ results in a new instance $I'$, which is similar to $I$ (the similarity between the two instances depends on the scale of the operation), then a natural and interesting problem arises: Is it easy to find a solution $S'$ to $I'$ that satisfies the specific quality guarantee, starting from the original solution $S$? In other words, how much runtime does a specific algorithm take to get a solution $S'$ to $I'$ with the quality guarantee, starting from $S$? The above setting is referred as the dynamic model of the given combinatorial optimization problem.

Studying the performances of evolutionary algorithms for dynamic models of combinatorial optimization problems is an emerging field in evolutionary computation [8, 16, 19, 22, 24, 26]. Within the paper, we present the dynamic model of the weighted vertex cover problem (WVC), which is simply named dynamic weighted vertex cover problem (DWVC). Our goal is to analyze the behaviors of the well-studied algorithms randomized local search (RLS) and $(1 + 1)$ EA that are adapted to DWVC. More specifically, we study the expected runtime (i.e., the expected number of fitness evaluations) that the algorithms need to recompute a 2-approximate solution when the given weighted graph is modified by a graph-editing operation, starting from a given 2-approximate solution to the original weighted graph. Note that all weighted graphs considered in the paper are vertex-weighted, i.e., the weight function is defined on the vertices, not edges.

Related work For the Vertex Cover problem, it is well-known that under the Unique Games Conjecture [14], there does not exist an approximation algorithm with a constant ratio $r < 2$, unless $P = \text{NP}$ [15]. The best-known 2-approximation algorithm for the Vertex Cover problem is based on the maximal matching: construct a maximal matching by greedily adding edges, then let the vertex cover contain both endpoints of each edge in the maximal matching. For WVC, Hochbaum [11] showed that a 2-approximate solution can be obtained by using the Linear Programming (LP) result of the fractional WVC. Du et al. [6] found that a maximal solution to the dual form [27] of the LP formulation (simply called dual formulation) for the fractional WVC also directly induces a 2-approximate solution. Using this conclusion, Bar-Yehuda and Even [2] presented a linear-time 2-approximation algorithm for WVC. The essential difference between the primal form of the LP formulation (simply called primal formulation) and the dual formulation for the fractional WVC is: the primal formulation considers the problem...
from the perspective of vertices; the dual formulation considers it from the perspective of edges [6]. (More details of the LP formulation and its dual formulation for the fractional WVC can be found in the next section.)

Pourhassan et al. [22] presented a dynamic model of the Vertex Cover problem, in which the graph-editing operation adds (resp., removes) exactly one edge into (resp., from) the given unweighted graph, and analyzed evolutionary algorithms with respect to their abilities to maintain a 2-approximate solution. They examined different variants of the RLS and (1 + 1) EA, node-based representation and edge-based representation. If using the node-based representation, they gave classes of instances for which both algorithms cannot get a 2-approximate solution in polynomial expected runtime with high probability. However, using the edge-based representation, they showed that the RLS and (1 + 1) EA can maintain 2-approximations efficiently if the algorithms start with a search point corresponding to a maximal matching of the original unweighted graph and use the fitness function given in [13] penalizing the edges sharing vertices.

Inspired by the work of Pourhassan et al. [22] and the essential difference between the primal and dual formulations of the fractional WVC, it is promising to consider DWVC from the perspective of edges, i.e., utilize the dual formulation to analyze DWVC. Here we give another example to show that using the dual formulation is better than using the primal formulation, to analyze DWVC. Consider a simplest graph-editing operation that removes or adds exactly one edge \([v, v']\). For the primal formulation, if a new edge \([v, v']\) is added into the graph, then the corresponding LP values of \(v\) and \(v'\) may be required to increase as their sum may be < 1 with respect to the given original LP solution; if an edge \([v, v']\) is removed from the graph, then the corresponding LP values of \(v\) and \(v'\) may have the room to decrease with respect to the given original LP solution. Thus there are two possible adjustment directions for the LP values of the vertices if using the primal formulation. For the dual formulation, because the given original maximal solution to the dual formulation does not violate the corresponding LP constraints no matter whether the edge \([v, v']\) is removed or added, so we only need to consider increasing the LP values of the edges. Therefore, using the dual formulation is able to simplify the analysis for DWVC, compared to using the primal formulation.

We formulate DWVC in the paper as: given a weighted graph \(G = (V, E, W)\) and a maximal solution \(S\) to the dual formulation of the fractional WVC on \(G\), the goal is to find a maximal solution to the dual formulation of the fractional WVC on the weighted graph \(G^* = (V^*, E^*, W^*)\) starting from the original maximal solution \(S\), where \(G^*\) is obtained by one of the two graph-editing operations on \(G\): (1) replace the edge-set \(E\) with a new one \(E^*\); (2) replace the weight function \(W\) with a new one \(W^*\). The version of DWVC with edge-modification is denoted by DWVC-E, and the one with weight-modification is denoted by DWVC-W. Denote by \(D \in \mathbb{N}^+\) the scale of the graph-editing operation, more specifically, \(D = |(E^* \setminus E) \cup (E \setminus E^*)|\) or \(D = |\{v \in V | W(v) \neq W^*(v)\}|\). It is necessary to point that both \(G\) and \(G^*\) are simple graphs (at most one edge between any two vertices).

Recently Pourhassan et al. [21] studied WVC using the dual formulation of the fractional WVC. As the weights on the vertices may be exponentially large with respect to the size of the graph (the number of edges), they incorporated the
Step Size Adaption strategy [3] into their (1 + 1) EA (Algorithm 4 given in [21]). However, their (1 + 1) EA was shown to take exponential expected runtime with high probability to get a maximal solution to the dual formulation. There are two factors causing the long runtime of their algorithm. Firstly, for a mutation $M$ constructed by their (1 + 1) EA, there may exist two edges selected by $M$ whose LP values are increased and decreased respectively. The randomness on the adjustment direction of the LP values leads that a mutation increases the sum of LP values for the edges with a relatively small probability, i.e., a mutation is rejected with a relatively large probability. Secondly, for a mutation $M$ that is rejected by their (1 + 1) EA, the step sizes of all the edges selected by $M$ would be decreased. Under the combined impact of the two factors, the step sizes of the edges cannot be increased enough to overcome the exponentially large weights on the vertices. That is, the step size adaption strategy is nearly invalid for their (1 + 1) EA.

Contributions Drawing on the experience of work [21] due to Pourhassan et al., we give two algorithms (1 + 1) EA and RLS adapted to DWVC with the step size adaption strategy as well. To avoid the invalidation of the step size adaption strategy that happens in the algorithm of [21], the two algorithms adopt an extra policy with three points: (1) the LP values of the edges selected by a mutation either all increase or all decrease (this only applies to the (1 + 1) EA, because any mutation of the RLS selects exactly one edge); (2) whether the algorithms increase or decrease the LP values of the edges depends on the fitness of the maintained solution; and (3) the condition to decrease the step size of a specific edge is very strict. Under the cooperation of the step size adaption strategy and the policy given above, the (1 + 1) EA and RLS are shown to take expected runtime $O\left(\frac{am}{\log_a W_{\text{max}} \cdot \log(max\{am, aD \cdot W_{\text{max}}\})} \right)$ to solve the two versions of DWVC (including two special variants for DWVC-E, and two special variants for DWVC-W), where $m$ denotes the number of edges in $G^*$, $W_{\text{max}} \geq 1$ denotes the maximum weight that the vertices in $G$ and $G^*$ have, and $a \in \mathbb{N}^+ \setminus \{1\}$ denotes the increasing/decreasing rate of the step size (i.e., the increment on the LP value for each edge can be exponentially increased or decreased by a factor $a$).

For the extra policy given above, its last two points play an important role in avoiding the invalidity of the step size adaption strategy, but they seem too restrictive and a little artificial. Thus we introduce the 1/5-th (success) rule, and give two algorithms with both the 1/5-th rule and step size adaption strategy, called the (1 + 1) EA with 1/5-th rule and RLS with 1/5-th rule. The 1/5-th rule is one of the best-known techniques in parameter control, especially in the control of the mutation probability (for more details, please refer to [4]). In the paper, we use the 1/5-th rule to control the increasing/decreasing rate of the step size. More specifically, the LP values of the edges selected by a mutation of the two algorithms with the 1/5-th rule increase or decrease with the same probability 1/2 (i.e., not depend on the maintained solution). If the mutation is accepted, then the step sizes of the selected edges are increased by a factor $a$; otherwise, decreased by a factor $a^{1/4}$. For the RLS with 1/5-th rule, we show that it can solve the two versions of DWVC (including the four special variants) efficiently. However, for the (1 + 1) EA with 1/5-th rule, we construct a special instance for each version.
of DWVC, and show that the algorithm takes at least pseudo-polynomial time to solve it. The main results given in the paper are summarized in Table 1.

The rest of the paper is structured as follows. We start by giving the related definitions and problem formulations in Sect. 2. Then we present the algorithms (1 + 1) EA and RLS (with the step size adaption strategy), and algorithms (1 + 1) EA with 1/5-th rule and RLS with 1/5-th rule for DWVC in two separated subsections of Sect. 3. For the two versions of DWVC, Sects. 4 and 5, respectively, analyze the expected runtime of the (1 + 1) EA and RLS, and the (1 + 1) EA with 1/5-th Rule and RLS with 1/5-th rule. Finally, conclusions are presented in Sect. 6.

2 Preliminaries

Consider a weighted graph $G = (V, E, W)$ with a vertex-set $V = \{v_1, \ldots, v_n\}$, an edge-set $E = \{e_1, \ldots, e_m\}$, and a weight function $W : V \rightarrow \mathbb{N}^+$ on the vertices. For any vertex $v \in V$, denote by $N_G(v)$ the set containing all the neighbors of $v$ in $G$, and by $E_G(v)$ the set containing all the edges incident to $v$ in $G$. For any vertex-subset $V' \subseteq V$, let $E_G(V') = \bigcup_{v \in V'} E_G(v)$. For any edge $e \in E$, denote by $E_G(e)$ the set containing all the edges in $G$ that have a common endpoint with $e$. For any edge-subset $E' \subseteq E$, let $E_G(E') = \bigcup_{e \in E'} E_G(e) \setminus E'$.

A vertex-subset $V_c \subseteq V$ is a vertex cover of $G$ if for each edge $e \in E$, either $e$ can be represented by its two endpoints $v$ and $v'$ as $[v, v']$, at least one of its two endpoints $v$ and $v'$ is in $V_c$. The weight of $V_c$ is defined as the sum of the weights on the vertices in $V_c$, written $\sum_{v \in V_c} W(v)$. The weighted vertex cover problem (WVC) on the weighted graph $G$ asks for a vertex cover of $G$ with the minimum weight, among all vertex covers of $G$.

Using the node-based representation (i.e. the search space is $\{0, 1\}^n$, and for any solution $x = x_1 \ldots x_n$ the node $v_i$ is chosen iff $x_i = 1$), the Integer Linear Programming (ILP) formulation for WVC is given as follows.

$$\begin{align*}
\min & \quad \sum_{i=1}^n W(v_i) \cdot x_i \\
\text{st.} & \quad x_i + x_j \geq 1 \quad \forall [v_i, v_j] \in E \\
& \quad x_i \in \{0, 1\} \quad i = 1, \ldots, n
\end{align*}$$

By relaxing the constraint $x_i \in \{0, 1\}$ of the ILP given above to $x_i \in [0, 1]$, the Linear Programming (LP) formulation for the fractional WVC is obtained. Hochbaum [11] showed that a 2-approximate solution can be found by using the LP result of the fractional WVC—include all the vertices $v_i$ with $x_i \geq 1/2$. The dual form of the LP formulation (or simply called dual formulation) for the fractional WVC is given as follows, where $Y : E \rightarrow \mathbb{R}^+ \cup \{0\}$ denotes a value assignment on the edges.

$$\begin{align*}
\max & \quad \sum_{e \in E} Y(e) \\
\text{st.} & \quad \sum_{e \in E_G(v)} Y(e) \leq W(v) \quad \forall v \in V
\end{align*}$$
Table 1 Overview on runtime performances of the four algorithms for the two versions of DWVC, DWVC-E and DWVC-W, including the two special variants for DWVC-E (DWVC-E* and DWVC-E+) and the two special variants for DWVC-W (DWVC-W* and DWVC-W+)

| Algorithm   | RLS or (1 + 1) EA | RLS with 1/5-th rule | (1 + 1) EA with 1/5-th rule |
|-------------|-------------------|----------------------|---------------------------|
| DWVC-E*     | \(O(am \log_a W_{\text{max}} \cdot \min(D, \log(aD \cdot \log_a W_{\text{max}})))\) | \(O(amd \log_a W_{\text{max}} \cdot \log W_{\text{max}})\) | \(\Omega(2^{\epsilon/2}), 0 < \epsilon \leq 1/2\) |
| DWVC-E      | \(O(am \log_a W_{\text{max}} \cdot \log(\max(am, aD \cdot W_{\text{max}})))\) | \(O(amd \log_a W_{\text{max}} \cdot \min(m \log W_{\text{max}} \cdot D \cdot W_{\text{max}}))\) | \(\Omega(2^{\epsilon/2}), 0 < \epsilon \leq 1/2\) |
| DWVC-W*     | \(O(am \log_a W_{\text{max}} \cdot \log(\max(am, aD \cdot W_{\text{max}})))\) | \(O(amd \log_a W_{\text{max}} \cdot \min(m \log W_{\text{max}} \cdot D \cdot W_{\text{max}}))\) | \(\Omega(2^{\epsilon/2}), 0 < \epsilon \leq 1/2\) |
| DWVC-W      | \(O(am \log_a W_{\text{max}} \cdot \log(\max(am, aD \cdot W_{\text{max}})))\) | \(O(amd \log_a W_{\text{max}} \cdot \min(m \log W_{\text{max}} \cdot D \cdot W_{\text{max}}))\) | \(\Omega(2^{\epsilon/2}), 0 < \epsilon \leq 1/2\) |

The notation \(m\) denotes the number of edges in the new graph, \(W_{\text{max}}\) denotes the maximum weight that the vertices in the original and new graphs have, \(D\) denotes the scale of the graph-editing operation, and \(a \in \mathbb{N}^+ \setminus \{1\}\) denotes the increasing/decreasing rate of the step size that is an integer ranging from 2 to \(W_{\text{max}}\). The lower bound of the runtime of the (1 + 1) EA with 1/5-th Rule for DWVC holds with probability \(1 - e^{-\epsilon/2}\) if \(W_{\text{max}} \geq a^m\).
The value assignment $Y$ is called a dual-solution of $G$ in the paper. Given a vertex $v \in V$, it satisfies the dual-LP constraint with respect to the dual-solution $Y$ if $\sum_{e \in E_G(v)} Y(e) \leq W(v)$. Similarly, for an edge $e \in E$, it satisfies the dual-LP constraint with respect to $Y$ if both its endpoints satisfy the dual-LP constraint with respect to $Y$. The dual-solution $Y$ of $G$ is feasible if all the vertices in $G$ satisfy the dual-LP constraint with respect to $Y$; otherwise, infeasible. The vertex $v \in V$ is tight with respect to $Y$ if $\sum_{e \in E_G(v)} Y(e) = W(v)$, and the edge $e \in E$ is tight with respect to $Y$ if at least one of its two endpoints is tight with respect to $Y$. Given a dual-solution $Y$ of $G$, denote by $V_G(Y)$ the set containing all the vertices in $G$ that do not satisfy the dual-LP constraint with respect to $Y$, and by $E_G(Y)$ the set containing all the edges that are incident to the vertices in $V_G(Y)$.

A maximal feasible dual-solution (MFDS) of $G$ is a feasible dual-solution of $G$ such that none of the edges can be assigned a larger LP value without violating the dual-LP constraint. Given an MFDS $Y$ of $G$, it induces a vertex cover of $G$ with ratio 2 directly, which contains all tight vertices with respect to $Y$ (a formal proof about the approximate ratio can be found in Theorem 8.4 of [6]).

Two versions of the dynamic weighted vertex cover problem (DWVC) are studied in the paper, whose formal formulations are given below.

**DWVC with Edge Modification (DWVC-E)**

*Input*: a weighted graph $G = (V, E, W)$, an MFDS $Y_{\text{orig}}$ of $G$, and a new edge-set $E^*$;

*Output*: an MFDS $Y^*$ of $G^* = (V, E^*, W)$

**DWVC with Weight Modification (DWVC-W)**

*Input*: a weighted graph $G = (V, E, W)$, an MFDS $Y_{\text{orig}}$ of $G$, and a new weight function $W^*$;

*Output*: an MFDS $Y^*$ of $G^* = (V, E, W^*)$

There are two special variants for DWVC-E, DWVC-E$^+$ and DWVC-E$^-$. Let $E^+ = E^* \setminus E$ and $E^- = E \setminus E^*$. The variants DWVC-E$^+$ and DWVC-E$^-$ consider the cases $E^- = \emptyset$ and $E^+ = \emptyset$, respectively. Similarly, there are two special variants for DWVC-W, DWVC-W$^+$ and DWVC-W$^-$. Let $V^+ = \{v \in V \mid W^*(v) > W(v)\}$ and $V^- = \{v \in V \mid W^*(v) < W(v)\}$. The variants DWVC-W$^+$ and DWVC-W$^-$ consider the cases $V^- = \emptyset$ and $V^+ = \emptyset$, respectively.

### 3 Four Adaptive Algorithms

We start with the subsection that introduces the $(1 + 1)$ EA and RLS, then the subsection that introduces the $(1 + 1)$ EA with 1/5-th rule and RLS with 1/5-th rule, finally the subsection that gives two simple lemmata based on the selection mechanism of the four algorithms.

It is worthy to point out that the solution maintained by the four algorithms is actually a $m$-dimensional vector (recall that $m$ is the number of edges in
$G^* = (V^*, E^*, W^*)$, \([x_1, x_2, \ldots, x_m]$, in which each \(x_i\) (\(1 \leq i \leq m\)) is the LP value of the edge \(e_i \in E^*\). We always use notation \(M\) to denote a mutation of the four algorithms, which corresponds to an adjustment (increment or decrement) on the values of some elements (corresponding to the edges chosen by \(M\)) in the vector.

### 3.1 (1+1) EA and RLS

Consider two weighted graphs $G = (V, E, W)$ and $G^* = (V^*, E^*, W^*)$, where $G^*$ is obtained by one of the two graph-editing operations mentioned above on $G$. We study the expected runtime (i.e., the expected number of fitness evaluations) of the (1+1) EA and RLS, given in Algorithm 1 and 2 respectively, to find an MFDS of $G^*$ starting with a given MFDS $Y_{\text{orig}}$ of $G$ (not from scratch). The two algorithms run in a similar way, except the mechanism selecting edges for mutation. The (1+1) EA selects each edge in $E^*$ with probability $1/m$ at each iteration, resulting in an edge-subset $I$ containing all the selected edges (see step 7 of Algorithm 1), and adjusts the LP values of the edges in $I$. The RLS differs from the (1+1) EA by selecting exactly one edge in $E^*$ in each round.

The two algorithms share the same general idea: If $Y_{\text{orig}}$ is also a feasible dual-solution of $G^*$, then they directly increase the LP values of the edges in $G^*$ until the LP value of any edge cannot be assigned with a larger value under the dual-LP constraint (i.e., an MFDS of $G^*$ is found if the claimed condition is met). Note that no infeasible dual-solution would be accepted during the process. If $Y_{\text{orig}}$ is an infeasible dual-solution of $G^*$, then the two algorithms first decrease the LP values of the edges in $E_{G^*}(Y_{\text{orig}})$ (because only the vertices in $V_{G^*}(Y_{\text{orig}})$ violate the dual-LP constraint with respect to $Y_{\text{orig}}$), aiming to get a feasible dual-solution $Y_t$ of $G^*$ as soon as possible, afterwards, increase the LP values of the edges in $G^*$ to get an MFDS based on $Y_t$.

The general idea of the two algorithms shows that the feasibility of the maintained solution decides the adjustment directions of the LP values of the selected edges. Thus we give a sign function $s(Y)$ below, to judge whether or not the considered solution $Y$ is a feasible dual-solution of $G^*$.

$$s(Y) = \begin{cases} -1 & \text{if } V_{G^*}(Y) \neq \emptyset, \text{ i.e., } Y \text{ is infeasible} \\ 1 & \text{feasible} \end{cases}$$

It is necessary to point out that a mutation of the (1+1) EA may choose more than one edge, and the LP values of the chosen edges are required to be either all increased or all decreased. In addition to the sign function $s()$, we also present a function $f(Y', Y)$ to compare the fitness of $Y'$ and $Y$, where $Y'$ is the dual-solution obtained by a mutation $M$ on the dual-solution $Y$ maintained by the two algorithms. It is defined as follows: $f(Y', Y) \geq 0$ if $Y'$ is not worse than $Y$; $f(Y', Y) < 0$ otherwise.

$$f(Y', Y) = \begin{cases} s(Y') \cdot \sum_{e \in E^*} (Y'(e) - Y(e)) & \text{if } s(Y) = 1 \\ \sum_{e \in E_{G^*}(Y)} (Y(e) - Y'(e)) - m \cdot W_{\max} \cdot \sum_{e \in E \setminus E_{G^*}(Y)} |Y(e) - Y'(e)| & \text{if } s(Y) = -1 \end{cases}$$
Algorithm 1: (1+1) EA

1. Initialize solution $Y$ and step size function $\sigma : E^* \rightarrow \{0,1\}$;
   
   // $Y(e) = Y_{orig}(e)$ for each $e \in E^* \cap E$, and $Y(e) = 0$ for each $e \in E^* \setminus E$

2. Determine $s(Y)$;

3. while the termination criteria not satisfied do

4. $Y' := Y$ and $I := \emptyset$; // set $I$ keeps all edges chosen by the mutation

5. for each edge $e \in E^*$ with probability $1/m$ do

6. $Y'(e) := \max \{ Y(e) + s(Y) \cdot \sigma(e), 0 \}$;

7. $I := I \cup \{ e \}$;

8. Determine $s(Y')$ and $f(Y', Y)$;

9. if $f(Y', Y) \geq 0$ then

10. $Y := Y'$;

11. $\sigma(e) := \min \{ \alpha \cdot \sigma(e), \alpha^{\log_2 W_{max} - 1} \}$ for all $e \in I$;

   // $\alpha$ is the increasing/decreasing rate of the step size

else

13. if $s(Y) > 0$ then

14. Let $I'$ be the subset of $I$ such that each edge $e \in I'$ violates the dual-LP constraint with respect to $Y'$, but no other edge in $I$ shares the endpoint that violates the dual-LP constraint with $e$;

15. $\sigma(e) := \max \{ \sigma(e)/\alpha, 1 \}$ for all $e \in I'$;

By the general idea of the two algorithms given above, if $Y$ is a feasible dual-solution of $G^*$, then the two algorithms increase the LP values of the edges, thus we always have that $\sum_{e \in E^*} Y'(e) \geq \sum_{e \in E^*} Y(e)$ for the obtained offspring $Y'$. If $Y'$ is infeasible, then $s(Y') = -1$ and $f(Y', Y) < 0$; otherwise, $s(Y') = 1$ and $f(Y', Y) \geq 0$. If $Y$ is an infeasible dual-solution of $G^*$, then the two algorithms decrease the LP values of the edges firstly, aiming to get a feasible dual-solution of $G^*$. Note that the LP values of the edges in $E^* \setminus E_{G^*}(Y)$ do not need to be decreased as they satisfy the dual-LP constraint with respect to $Y$. If they are decreased during the process to get the first feasible dual-solution, then the algorithm may spend much extra time to make up the decrements on the LP values of the edges in $E^* \setminus E_{G^*}(Y)$ (i.e., spend extra time to make the edges in $E^* \setminus E_{G^*}(Y)$ be tight again). Thus the term of $f(Y', Y)$,

$$-m \cdot W_{max} \cdot \sum_{e \in E^* \setminus E_{G^*}(Y)} |Y(e) - Y'(e)|,$$
penalizes the mutation that decreases the LP values of the edges in $E^* \setminus C(Y)$, which guides the mutation to decrease only the LP values of the edges in $E_G(Y)$. More specifically, if the LP value of some edge in $E^* \setminus C(Y)$ is changed by the considered mutation (note that the increment or decrement on the LP value is always $\geq 1$), then

$$-m \cdot W_{\text{max}} \cdot \sum_{e \in E^* \setminus C(Y)} |Y(e) - Y'(e)| \leq -m \cdot W_{\text{max}}.$$  \hspace{1cm} (1)

Now we consider the upper bound of the term $\sum_{e \in E_G(Y)} (Y(e) - Y'(e))$ under the assumption that the LP value of some edge in $E^* \setminus C(Y)$ is decreased by the considered mutation. Since all solutions obtained during the process from the initial solution $Y_{\text{orig}}$ to $Y$ are infeasible (including $Y_{\text{orig}}$), $Y(e) \leq Y_{\text{orig}}(e) \leq W_{\text{max}}$ for each edge $e \in E^*$, i.e., $0 \leq Y(e) - Y'(e) \leq W_{\text{max}}$. Moreover, as $E^* \setminus C(Y)$ cannot be empty, $|E_G(Y)| < m$. Therefore,

$$\sum_{e \in E_G(Y)} (Y(e) - Y'(e)) < m \cdot W_{\text{max}}.$$  \hspace{1cm} (2)

Combining Inequalities 1 and 2, $f(Y', Y) < 0$ no matter whether $Y'$ is feasible or infeasible, implying that the mutation would be rejected if it changes the LP value of some edge in $E^* \setminus C(Y)$. For the case that no LP value of the edges in $E^* \setminus C(Y)$ is decreased by the considered mutation, it is easy to see that $f(Y', Y) \geq 0$.

\begin{algorithm}
\caption{RLS}
\begin{algorithmic}[1]
\State Initialize solution $Y$ and step size function $\sigma : E^* \to 1$ ;
  \Comment{$Y(e) = Y_{\text{orig}}(e)$ for each $e \in E^* \cap E$, and $Y(e) = 0$ for each $e \in E^* \setminus E$}
\State Determine $s(Y)$ ;
\While{the termination criteria not satisfied}
\State $Y' := Y$ ;
\State Choose an edge $e \in E^*$ uniformly at random ;
\State $Y'(e) := \max\{Y(e) + \sigma(e) \cdot s(Y), 0\}$ ;
\State Determine $s(Y')$ and $f(Y', Y)$ ;
\If{$f(Y', Y) \geq 0$}
\State $Y := Y'$ and $\sigma(e) := \min\{\alpha \cdot \sigma(e), \alpha^{\lceil \log_\alpha W_{\text{max}} \rceil + 1}\}$ ;
  \Comment{$\alpha$ is the increasing/decreasing rate of the step size}
\Else
\State \If{$s(Y) > 0$}
\State $\sigma(e) := \max\{\sigma(e)/\alpha, 1\}$ ;
\EndIf
\EndIf
\EndWhile
\end{algorithmic}
\end{algorithm}
To deal with the case that the weights on the vertices are exponentially large with respect to the size of the graph (the number $m$ of edges), the Step Size Adaptation strategy [3] is incorporated into the two algorithms (see steps 9–15 of Algorithm 1 and steps 8–12 of Algorithm 2): the increment (called step size in the following text) on the LP values of the edges can exponentially increase or decrease. Let $\sigma : E^* \to \mathbb{N}^+$ be the step size function that keeps the step size for each edge in $E^*$, and let $\sigma$ be initialized as $\sigma : E^* \to 1$.

Given a mutation of the RLS on $Y$, if it is accepted (i.e., $f(Y', Y) \geq 0$, where $Y'$ is the solution obtained by the mutation on $Y$), then the step size of the chosen edge $e$ is increased by a factor $\alpha$, where $\alpha$ is an integer between 2 and $W_{\text{max}}$; otherwise, decreased by a factor $\alpha$ if $s(Y) > 0$. W.l.o.g., we assume that the step size of each edge can be upper and lower bounded by $\alpha^\lfloor \log_\alpha W_{\text{max}} \rfloor + 1$ and 1, respectively. Given a mutation of the (1 + 1) EA on $Y$, if it is accepted, then the step size of each edge $e \in I$ is increased by a factor $\alpha$ if $s(Y) > 0$, where $I'$ is the subset of $I$ such that each edge $e \in I'$ violates the dual-LP constraint with respect to $Y'$, but no other edge in $I$ shares the endpoint that violates the dual-LP constraint with $e$ (see step 14 of Algorithm 1). The reason why we define the subset $I'$ of $I$ is that we can ensure that the step size of each edge in $I'$ is unfit for $Y$. For an edge $e$ in $I \setminus I'$, there are two cases: (1) neither its two endpoints violates the dual-LP constraint with respect to the dual-solution $Y'$; (2) there is another edge $e' \in I \setminus \{e\}$ that has a common endpoint with $e$ such that the common endpoint of $e$ and $e'$ violates the dual-LP constraint with respect to the dual-solution $Y'$. For case (1), we should not decrease its step size. For case (2), we cannot conclude that the step size of $e$ is unfit for the solution $Y$, because the step size of $e$ may be fit for $Y$ if it is considered independently. If the algorithms adopt a “radical” strategy that decreases the step sizes of all the edges in $I$ if the mutation is rejected, then they would spend much time on increasing the step sizes of the edges (in some extreme case, the step size cannot exponentially increase, resulting in an exponential waiting time to get an MFDS [21]). Thus we adopt a “conservative” strategy: Only decrease the step sizes of the edges in $I'$.

Note that for any mutation of the (1 + 1) EA or RLS that is rejected, the step sizes of the edges selected by the mutation are not decreased if $s(Y) < 0$, because the rejection of the mutation is caused by the selection of the edges, not the violation of the dual-LP constraint.

### 3.2 (1 + 1) EA with 1/5-th Rule and RLS with 1/5-th Rule

To eliminate the artificial influences on the two algorithms given in the previous subsection, such as the adjustment direction (increasing or decreasing the LP values) controlled by the sign function $s()$, and the strict condition to decrease the step size of a specific edge given in the (1 + 1) EA (only the step sizes of the edges in $I'$ can be decreased if the mutation is rejected and $s() > 0$), we incorporate the 1/5-th (success) rule, and present two algorithms, the (1 + 1) EA with 1/5-th rule and RLS with 1/5-th rule, given in Algorithm 3 and 4 respectively. The two algorithms follow the fitness comparing function $f(Y', Y)$ defined in the previous subsection.
The general idea of the two algorithms is: no matter whether or not the current maintained dual-solution is feasible, they either increase or decrease the LP values of the edges selected by the mutation with the same probability 1/2 (depend on the value of $b$, see step 4 of Algorithm 3 and step 5 of Algorithm 4). If the mutation is accepted, then the dual-solution is updated, and the step sizes of these chosen edges are increased by a factor $\alpha^{b \cdot \sigma(e)}$, otherwise, the step sizes of these chosen edges are decreased by a factor $\alpha^{-b \cdot \sigma(e)}$. It is necessary to point out that for a mutation of the $(1+1)$ EA with $1/5$-th rule, we still require that the LP values of the edges selected by the mutation either all increase or all decrease.

The previous subsection analyzed the cases of the fitness comparing function $f(Y', Y)$ when the LP values of the edges are increased if $s(Y) = 1$, or the LP values of the edges are decreased if $s(Y) = -1$. Here we supplement the analysis of the case that the LP values of the edges are decreased if $s(Y) = -1$, and the case that the LP values of the edges are increased if $s(Y) = 1$. If $s(Y) = 1$ and the LP values of the edges are decreased, then obviously $s(Y') = 1$, and

$$f(Y', Y) = \sum_{e \in E^*} (Y'(e) - Y(e)) < 0.$$  

If $s(Y) = -1$ and the LP values of some edges are increased, then
3.3 Observations Based on Fitness Comparing Function

The selection mechanism of the four algorithms given above implies the following two lemmata.

**Lemma 1** Given two dual-solutions $Y$ and $Y'$, where $Y'$ is obtained by a mutation of the four algorithms on $Y$, if $Y'$ is accepted then $s(Y') \geq s(Y)$.

**Lemma 2** Given two dual-solutions $Y$ and $Y'$, where $Y'$ is obtained by a mutation of the four algorithms on $Y$, if $Y$ is infeasible and $Y'$ is accepted, then the mutation only decreases the LP values of the edges in $E_{G^*}(Y)$.

$$f(Y', Y) = \sum_{e \in E_{G^*}(Y)} (Y(e) - Y'(e)) - m \cdot W_{\max} \cdot \sum_{e \in E \setminus E_{G^*}(Y)} |Y(e) - Y'(e)| < 0.$$
4 Runtime Analysis for the (1 + 1) EA and RLS

We start the section with a notion related to mutation, which plays an important role in the following discussion. Given an edge \( e \) in the weighted graph \( G^* \), a mutation of the (1 + 1) EA or RLS is a \textit{valid mutation on} \( e \) if it results in an increment or decrement on the LP value of \( e \), or on the step size \( \sigma(e) \) of \( e \). Note that if the mutation is of the (1 + 1) EA, then it may choose some other edges in addition to \( e \). The two lemmata given below study the behaviors of the (1 + 1) EA and RLS on a specific edge \( e^* = [v_1, v_2] \) in \( G^* \).

\begin{lemma}
Consider a feasible dual-solution \( Y^* \) of \( G^* \), and an initial value \( \sigma_I \) of the step size of the edge \( e^* \). For a feasible dual-solution \( Y^* \) obtained by the (1 + 1) EA (or RLS) starting with \( Y^* \), where \( Y^*(e^*) - Y^*(e^*) \geq \sigma_I \), the algorithm takes expected runtime \( \Omega(am \log_a (Y^*(e^*) - Y^*(e^*)) \) to increase the LP value of \( e^* \) from \( Y^*(e^*) \) to \( Y^*(e^*) \).
\end{lemma}

\textbf{Proof} We start with the analysis for the (1 + 1) EA. Since \( Y^* \) is a feasible dual-solution of \( G^* \), by Lemma 1, the sign function \( s() \) remains at 1 during the process from \( Y^* \) to \( Y^* \), indicating that the LP value of \( e^* \) is monotonically increased from \( Y^*(e^*) \) to \( Y^*(e^*) \). Let \( Y \) be an arbitrary accepted solution of the (1 + 1) EA during the process, \( M \) be a mutation of the (1 + 1) EA on \( Y \), and \( Y' \) be the offspring obtained by \( M \) on \( Y \). In the following discussion, we first analyze the impact of the mutation \( M \) on the step size \( \sigma(e^*) \) of \( e^* \), where the notation \( \sigma(e^*) \) here denotes the step size of \( e^* \) before the generation of \( M \). Observe that \( M \) cannot influence \( \sigma(e^*) \) if \( e^* \notin I \), where \( I \) denotes the set containing all the edges selected by \( M \) (see step 7 of Algorithm 1). Thus in the following discussion, we assume that \( e^* \in I \).

Case (1). \( \sigma(e^*) \leq Y^*(e^*) - Y(e^*) \). If \( M \) is accepted by the (1 + 1) EA, then the step size of \( e^* \) is increased from \( \sigma(e^*) \) to \( \alpha \cdot \sigma(e^*) \); otherwise, the analysis on \( M \) is divided into the two subcases given below.

Case (1.1). An endpoint \( v_1 \) of \( e^* \) violates the dual-LP constraint with respect to \( Y' \). Since \( \sigma(e^*) \leq Y^*(e^*) - Y(e^*) \), the edge-subset \( (E_G(v_1) \cap I) \setminus \{e^* \} \) cannot be empty, and the increments on the LP values of the edges in \( E_G(v_1) \cap I \) results in the dual-LP constraint violation of \( v_1 \) with respect to \( Y' \). According to the definition of the edge-set \( I' \) (see step 14 of Algorithm 1), we have that \( e^* \notin I' \), and \( M \) cannot influence \( \sigma(e^*) \).

Case (1.2). No endpoint of \( e^* \) violates the dual-LP constraint with respect to \( Y' \). According to the definition of the edge-set \( I' \), we also have that \( e^* \notin I' \), and \( M \) cannot influence \( \sigma(e^*) \). By the above analysis, any mutation of the (1 + 1) EA cannot cause an decrement on the step size of \( e^* \) under Case (1). If the mutation \( M \) only selects the edge \( e^* \), then it is a valid mutation on \( e^* \), and can be accepted by the algorithm. The (1 + 1) EA generates such a valid mutation on \( e^* \) with probability \( \Omega(1/m) \). Thus under Case (1), the algorithm takes expected runtime \( \Omega(m) \) to increase the LP value of edge \( e^* \) from \( Y(e^*) \) to \( Y(e^*) + \sigma(e^*) \), and increase the step size of \( e^* \) from \( \sigma(e^*) \) to \( \alpha \cdot \sigma(e^*) \).
Case (2). $\sigma(e^*) > Y^\dagger(e^*) - Y(e^*)$. For the case, the mutation $M$ would be rejected by the $(1 + 1)$ EA as $e^* \in I$. The analysis on $M$ can be divided into the following two subcases.

Case (2.1). There is no edge in $I \setminus \{e^*\}$ sharing the endpoint of $e^*$ that violates the dual-LP constraint with respect to $Y'$. For this subcase, $e^* \in I'$, and the step size of $e^*$ is decreased from $\sigma(e^*)$ to $\sigma(e^*)/\alpha$.

Case (2.2). There is an edge $e_1^* \in I \setminus \{e^*\}$ sharing the endpoint of $e^*$ that violates the dual-LP constraint with respect to $Y'$. Because of the existence of $e_1^*$, $e^* \notin I'$ and $M$ does not influence the step size of $e^*$.

If the mutation $M$ only selects the edge $e^*$, then it is valid mutation on $e^*$, and belongs to Case (2.1). The $(1 + 1)$ EA generates such a valid mutation on $e^*$ with probability $\Omega(1/m)$. Thus under Case (2), the $(1 + 1)$ EA takes expected runtime $O(m)$ to decrease the step size of $e^*$ from $\sigma(e^*)$ to $\sigma(e^*)/\alpha$.

Now we are ready to analyze the expected runtime of the $(1 + 1)$ EA to increase the LP value of $e^*$ from $Y^\dagger(e^*)$ to $Y^\dagger(e^*)$, using the above obtained results. Since $Y^\dagger(e^*) - Y^\dagger(e^*) \geq \sigma_1$, the whole process can be divided into Phase (I) and Phase (II). Phase (I) contains all steps of the algorithm until the step size of $e^*$ is decreased for the first time, i.e., the step size of $e^*$ can only increase during the phase. More specifically, the condition of Case (1) is always met with respect to the maintained solution $Y$ during Phase (I). Phase (II) follows Phase (I), during which the step size of $e^*$ may increase or decrease, but the general trend is decreasing. W.l.o.g., assume that the initial value $\sigma_1$ of the step size of $e^*$ is equal to $a^p$, where $p \geq 0$ is an integer not less than 0.

**Phase (I)** Let $q$ be the integer such that

$$\sum_{i=p}^{q} \alpha^i \leq Y^\dagger(e^*) - Y^\dagger(e^*) \quad \text{and} \quad \sum_{i=p}^{q+1} \alpha^i > Y^\dagger(e^*) - Y^\dagger(e^*).$$

Now it is easy to see that the step size of $e^*$ can be increased from $a^p$ to $a^{q+1}$ during the phase. Thus the number of valid mutations on $e^*$ required during Phase (I) is $q - p + 1$, where

$$q - p + 1 = \left\lceil \log_a \left( \frac{Y^\dagger(e^*) - Y^\dagger(e^*)}{a^p} \right) + 1 \right\rceil.$$ 

Combining the expected runtime of the algorithm to generate a valid mutation on $e^*$, Phase (I) takes expected runtime $O(m \log_a \left( Y^\dagger(e^*) - Y^\dagger(e^*) \right))$ (because $p$ may be 0).

**Phase (II)** During the phase, the LP value of $e^*$ is increased from $Y^\dagger(e^*) + \sum_{i=p}^{q} \alpha^i$ to $Y^\dagger(e^*)$, and the step size of $e^*$ is decreased from $a^{q+1}$ to 1. Similar to the analysis for Phase (I), we analyze the number $T$ of valid mutations on $e^*$ during Phase (II). However, to simplify the analysis, we separately consider the number $t_i$ of valid mutations on $e^*$ with step size $a^i$ among the $T$ valid mutations on $e^*$, where $0 \leq i \leq q + 1$ (since the step size of $e^*$ can increase or decrease during Phase (II), there may be more than one valid mutation on $e^*$ with step size $a^i$). Obviously $T = \sum_{i=0}^{q+1} t_i$.
We start with the analysis for $t_{q+1}$. Since the valid mutation on $e^*$ with step size $a^{q+1}$ cannot be accepted, the step size will be decreased to $a^q$. However, if a valid mutation on $e^*$ with step size $a^q$ is accepted, then the step size will be increased to $a^{q+1}$ again. Thus $t_{q+1} \leq 1 + (\alpha - 1) = \alpha$, because there are at most $\alpha - 1$ valid mutations on $e^*$ with step size $a^q$ among the $T$ valid mutations on $e^*$ that can be accepted by the algorithm. Now we consider $t_i$ for any $1 \leq i \leq q$, under the assumption that the mutation on $e^*$ with step size $a^{i+1}$ cannot be accepted. Using the reasoning similar to that given above for the mutation on $e^*$ with step size $a^{i+1}$, we can get that there are at most $\alpha$ valid mutations on $e^*$ with step size $a^i$ that can be rejected among the $T$ valid mutations on $e^*$. Combining it with the observation that there are at most $\alpha - 1$ valid mutations on $e^*$ with step size $a^i$ that can be accepted, we can derive that $t_i \leq 2\alpha - 1$. Once the step size of $e^*$ is decreased to 1, then the LP value of $e^*$ is between $Y^\dagger(e^*) - \alpha + 1$ and $Y^\dagger(e^*)$. If the LP value of $e^*$ equals $Y^\dagger(e^*)$, then Phase (II) is over, and $t_0 = 0$. If the LP value of $e^*$ is between $Y^\dagger(e^*) - \alpha + 1$ and $Y^\dagger(e^*) - 1$, then $t_0 \leq \alpha - 1$. The above analysis gives

$$T = \sum_{i=0}^{q+1} t_i \leq (2\alpha - 1) \cdot (q + 1).$$

By the analysis for Case (1–2), Phase (II) takes expected runtime $O(\alpha m \log_a (Y^\dagger(e^*) - Y^\dagger(e^*)).$ Summarizing the above analysis for the two phases, there are at most $2\alpha(q + 1)$ valid mutations on $e^*$ during the process from $Y^\dagger$ to $Y^\ddagger$, for which the $(1 + 1)$ EA takes expected runtime $O(\alpha m \log_a (Y^\dagger(e^*) - Y^\dagger(e^*)).$ Since the RLS chooses exactly one edge in each iteration, any mutation of the RLS on $e^*$ is valid. Using the reasoning similar to that given above, we can get the same expected runtime for the RLS.

Now we analyze the expected runtime of the two algorithms to make the edge $e^*$ satisfy the dual-LP constraint, if they start with an infeasible dual-solution with respect to which $e^*$ violates the dual-LP constraint.

**Lemma 4** Consider an infeasible dual-solution $Y^\dagger$ of $G^*$, with respect to which the edge $e^*$ violates the dual-LP constraint. For the first feasible dual-solution $Y^\ddagger$ obtained by the $(1 + 1)$ EA (or RLS) starting with $Y^\dagger$, the algorithm takes expected runtime $O(m \log_a (Y^\dagger(e^*) - Y^\ddagger(e^*)))$ to decrease the LP value of $e^*$ from $Y^\dagger(e^*)$ to $Y^\ddagger(e^*)$.

**Proof** We start with the analysis for the $(1 + 1)$ EA. Since $Y^\ddagger$ is the first feasible dual-solution obtained by the $(1 + 1)$ EA starting with $Y^\dagger$, the LP value of $e^*$ is monotonically decreased from $Y^\dagger(e^*)$ to $Y^\ddagger(e^*)$. Assume that the step size of $e^*$ is initialized as $a^p$, where $p \geq 0$ is an integer not less than 0. Observe that the step size of $e^*$ cannot decrease during the process from $Y^\dagger$ to $Y^\ddagger$ because the sign function remains at $-1$. Hence if $Y^\ddagger(e^*) > 0$, then there exists an integer $q$ such that $\sum_{i=p}^{q} a^i = Y^\dagger(e^*) - Y^\ddagger(e^*)$, and the step size of $e^*$ is
increased from $a^p$ to $a^{q+1}$ during the process. Consequently, the process contains $q - p + 1$ valid mutations on $e^*$, where

$$q - p + 1 = \log_a \left( \frac{(Y^\dagger(e^*) - Y^\ddagger(e^*)) \cdot (a - 1)}{a^p} + 1 \right).$$

If $Y^\ddagger(e^*) = 0$, then there exists an integer $q$ such that $\sum_{i=p}^{q-1} a^i < Y^\dagger(e^*)$, $\sum_{i=p}^{q} a^i \geq Y^\dagger(e^*)$, and the step size of $e^*$ is increased from $a^p$ to $a^{q+1}$ during the process. Similarly, the process contains $q - p + 1$ valid mutations on $e^*$, where

$$q - p + 1 = \left\lfloor \log_a \left( \frac{(a - 1) \cdot Y^\dagger(e^*)}{a^p} + 1 \right) \right\rfloor.$$

The mutation that only selects the edge $e^*$ is a valid mutation on $e^*$, which can be generated by the $(1 + 1)$ EA with probability $\Omega(1/m)$. Thus the $(1 + 1)$ EA takes expected runtime $O(m(q + 1)) = O(m \log_a (Y^\dagger(e^*) - Y^\ddagger(e^*)))$ to get $Y^\ddagger$ (because $p$ may be 0). The above conclusions for the $(1 + 1)$ EA also apply to the RLS. \qed

### 4.1 Analysis for DWVC with Edge Modification

We start the subsection with the analysis of the algorithms $(1 + 1)$ EA and RLS for the two special variants of DWVC-E, namely, DWVC-E⁺ and DWVC-E⁻. Denote by $E^+ = E^* \setminus E$ the set containing all the new added edges, and by $E^- = E \setminus E^*$ the set containing all the removed edges. The variant DWVC-E⁺ considers the case that $E^- = \emptyset$, and DWVC-E⁻ considers the case that $E^+ = \emptyset$.

The following theorem analyzes the performances of the two algorithms for DWVC-E⁺, from two different views. We remark that for an instance $\{G = (V, E, W), Y_{\text{orig}}, E^+\}$ of DWVC-E⁺, $|E^+| = D$, and for each edge $e \in E^+$, $Y_{\text{orig}}(e)$ and $\sigma(e)$ are initialized as 0 and 1, respectively.

**Theorem 5** The expected runtime of the $(1 + 1)$ EA (or RLS) for DWVC-E⁺ is $O(am \log_a W_{\text{max}} \cdot \min\{D, \log(aD \cdot \log_a W_{\text{max}})\})$.

**Proof** We first consider the expected runtime of the $(1 + 1)$ EA to obtain an MFDS of $G^* = (V, E \cup E^+, W)$, starting with the given MFDS $Y_{\text{orig}}$ of $G = (V, E, W)$. Observe that $Y_{\text{orig}}$ is a feasible dual-solution of $G^*$. Thus combining Lemma 1 and the general idea of the algorithm, we have that all mutations accepted by the algorithm increase the LP values of the edges in $G^*$. If $Y_{\text{orig}}$ is an MFDS of $G^*$, then any mutation on $Y_{\text{orig}}$ results in an infeasible solution that would be rejected, i.e., the algorithm keeps the dual-solution $Y_{\text{orig}}$ forever.

In the following discussion, we assume that $Y_{\text{orig}}$ is not an MFDS of $G^*$. Observe that any increment on the LP values of the edges in $E$ would result in an infeasible solution that cannot be accepted by the algorithm. Thus we have that
$Y^*(e) = Y_{\text{orig}}(e)$ for each edge $e \in E$, and $Y^*(e) \geq Y_{\text{orig}}(e)$ for each edge $e \in E^+$, where $Y^*$ is an MFDS of $G^*$ obtained by the $(1 + 1)$ EA starting with $Y_{\text{orig}}$. To study the expected runtime of the $(1 + 1)$ EA to get $Y^*$, two analytical ways from different views are given below: One considers the edges in $E^+$ sequentially; the other one considers that in an interleaved way.

We start with the analysis from the view that considers the edges in $E^+$ sequentially. Let $e^* = [v_1, v_2]$ be an arbitrary edge in $E^+$ with $Y^*(e^*) - Y_{\text{orig}}(e^*) > 0$. Since $Y^*(e^*) - Y_{\text{orig}}(e^*) \leq W_{\text{max}}$ and the fact that the step size of $e^*$ is initialized with value 1, Lemma 3 gives that the $(1 + 1)$ EA takes expected runtime

$$O(am \log_\alpha (Y^*(e^*) - Y_{\text{orig}}(e^*))) = O(am \log_\alpha W_{\text{max}})$$

to increase the LP value of $e^*$ from $Y_{\text{orig}}(e^*)$ to $Y^*(e^*)$. Combining the fact that the number of edges in $E^+$ is bounded by $D$, we have that the $(1 + 1)$ EA takes expected runtime $O(amD \log_\alpha W_{\text{max}})$ to get $Y^*$.

Now we analyze the expected runtime of the $(1 + 1)$ EA to get $Y^*$ from the other view that considers the edges in $E^+$ as a whole. For each edge $e \in E^+$, denote $Y^*(e) - Y_{\text{orig}}(e)$ by $\Delta(e)$, and denote by $\beta(e)$ the number of valid mutations on $e$ that the algorithm requires to increase the LP value of $e$ from $Y_{\text{orig}}(e)$ to $Y^*(e)$. Let $E_\Delta = \{e \in E^+| \Delta(e) \neq 0\}$, and let the potential of the dual-solution $Y_{\text{orig}}$ be

$$g(Y_{\text{orig}}) = \sum_{e \in E_\Delta} \beta(e).$$

(3)

Observe that $E_\Delta \subset E^+$. Since there may exist a mutation that is not only a valid mutation on $e_1 \in E^+$, but also a valid mutation on $e_2 \in E^+ \setminus \{e_1\}$, $g(Y_{\text{orig}})$ is the upper bound of the number of valid mutations on the edges in $E_\Delta$ that the algorithm requires to get $Y^*$ starting from $Y_{\text{orig}}$. Moreover, the analysis of Lemma 3 gives that any mutation on $Y_{\text{orig}}$ cannot increase its potential.

To obtain the expected drift of $g$, we first consider the relation between $|E_\Delta|$ and $g(Y_{\text{orig}})$. For each edge $e \in E_\Delta$, Lemma 3 gives that

$$\beta(e) \leq 2\alpha \lfloor \log_\alpha ((\alpha - 1) \cdot \Delta(e) + 1) \rfloor \leq 2\alpha \log_\alpha (\alpha \cdot \Delta(e)) \leq 2\alpha (\log_\alpha W_{\text{max}} + 1).$$

(4)

By Eqs. 3 and 4, we have that

$$|E_\Delta| \geq \frac{g(Y_{\text{orig}})}{2\alpha (\log_\alpha W_{\text{max}} + 1)} \quad \text{and} \quad g(Y_{\text{orig}}) \leq D \cdot 2\alpha (\log_\alpha W_{\text{max}} + 1).$$

A valid mutation that chooses exactly one of the edge in $E_\Delta$ can be generated by the algorithm with probability $\Omega(|E_\Delta|/(e \cdot m))$, which results in a new solution $Y'$ with $g(Y') = g(Y_{\text{orig}}) - 1$. Thus the expected drift of $g$ can be bounded by

$$\frac{|E_\Delta|}{e \cdot m} \geq \frac{g(Y_{\text{orig}})}{e \cdot 2\alpha m \cdot (\log_\alpha W_{\text{max}} + 1)}.$$
As mentioned above, the maximum value that \( g(Y_{\text{orig}}) \) can take is 
\[ 2aD(\log_{a} W_{\text{max}} + 1). \]
Combining it, the obvious minimum value 1 that \( g(Y_{\text{orig}}) \) can take, and the expected drift of \( g \), the Multiplicative Drift Theorem [5] gives that the (1 + 1) EA takes expected runtime \( O(\alpha m \log_{a} W_{\text{max}} \cdot \log(aD \cdot \log_{a} W_{\text{max}})) \) to get \( Y^* \).

Summarizing the above analysis, we can conclude that the (1 + 1) EA takes expected runtime \( O(\alpha m \log_{a} W_{\text{max}} \cdot \min\{D, \log(aD \cdot \log_{a} W_{\text{max}})\}) \) to find an MFDS of \( G^* \) starting with \( Y_{\text{orig}} \). Since we only consider the mutations selecting exactly one edge in the analysis for the (1 + 1) EA, the above conclusions also apply to the RLS.

Given an instance \( \{G = (V, E, W), Y_{\text{orig}}, E^-\} \) of DWVC-E\(^-\), the following theorem considers the expected runtime of the (1 + 1) EA and RLS to obtain an MFDS of \( G^* = (V, E \setminus E^-, W) \), starting with the MFDS \( Y_{\text{orig}} \) of \( G \). Note that the domain of definition for \( Y_{\text{orig}} \) and the weight function \( W \) are modified as \( E \setminus E^- \) after the removal, and \(|E^-| = D\).

**Theorem 6** The expected runtime of the (1 + 1) EA (or RLS) for DWVC-E\(^-\) is \( O(\alpha m \log_{a} W_{\text{max}} \cdot \log(\max\{\alpha m, aD \cdot W_{\text{max}}\})) \).

**Proof** Observe that \( Y_{\text{orig}} \) is a feasible dual-solution of \( G^* \), and the endpoints of the edges in \( E^- \) may not be tight with respect to \( Y_{\text{orig}} \) once the edges in \( E^- \) are removed. Thus the LP values of the edges in \( E_G(E^-) \) may have the room to be increased. If \( Y_{\text{orig}} \) is an MFDS of \( G^* \), then any mutation of the (1 + 1) EA (or RLS) on \( Y_{\text{orig}} \) would be rejected, and the algorithm keeps the dual-solution \( Y_{\text{orig}} \) forever. In the following discussion, we assume that \( Y_{\text{orig}} \) is not an MFDS of \( G^* \).

Let \( Y^* \) be an arbitrary MFDS of \( G^* \) obtained by the (1 + 1) EA (or RLS) starting with \( Y_{\text{orig}} \). The above analysis gives that \( Y^*(e) = Y_{\text{orig}}(e) \) for each edge \( e \in E \setminus (E^- \cup E_G(E^-)) \), and \( Y^*(e) \geq Y_{\text{orig}}(e) \) for each edge \( e \in E_G(E^-) \). Observe that all the edges in \( E_G(E^-) \) are incident to the endpoints of the edges in \( E^- \), and the number of endpoints of the edges in \( E^- \) is upper bounded by \( 2D \). Combining the observation with the fact that the sum of the LP values of the edges sharing an endpoint cannot be larger than the weight of the endpoint under the dual-LP constraint, we have that \( \sum_{e \in E_G(E^-)} Y^*(e) \) can be upper bounded by \( 2D \cdot W_{\text{max}} \).

For each edge \( e \in E_G(E^-) \), denote \( Y^*(e) - Y_{\text{orig}}(e) \) by \( \Delta(e) \), and denote by \( \beta(e) \) the number of valid mutations on \( e \) that the algorithm requires to increase the LP value of \( e \) from \( Y_{\text{orig}}(e) \) to \( Y^*(e) \). Let \( E_\Delta = \{e \in E_G(E^-) | \Delta(e) \neq 0\} \). Then we have

\[
\sum_{e \in E_\Delta} \Delta(e) = \sum_{e \in E_G(E^-)} (Y^*(e) - Y_{\text{orig}}(e)) \leq \sum_{e \in E_G(E^-)} Y^*(e) \leq 2D \cdot W_{\text{max}}.
\]

Let the potential of the solution \( Y_{\text{orig}} \) be

\[
g(Y_{\text{orig}}) = \sum_{e \in E_\Delta} \beta(e).
\]

Similar to the analysis given in Theorem 5, we have that \( g(Y_{\text{orig}}) \) is the upper bound of the number of valid mutations on the edges in \( E_\Delta \) that the algorithm requires to
get $Y^*$ starting from $Y_{\text{orig}}$, and any mutation on $Y_{\text{orig}}$ cannot increase its potential. The analysis for the expected drift of $g$ is divided into two cases, based on the value of $\sum_{e \in E_\Delta} \Delta(e) / |E_\Delta|$

Case (1). $\sum_{e \in E_\Delta} \Delta(e) < \alpha \cdot |E_\Delta|$. Lemma 3 gives that $\beta(e) \leq 2\alpha (\log_\alpha \Delta(e) + 1)$ for each edge $e \in E_\Delta$. Thus we have

$$g(Y_{\text{orig}}) = \sum_{e \in E_\Delta} \beta(e) \leq 2\alpha \cdot |E_\Delta| + 2\alpha \cdot \log_\alpha \left( \prod_{e \in E_\Delta} \Delta(e) \right)$$

$$\leq 2\alpha \cdot |E_\Delta| + 2\alpha \cdot |E_\Delta| \cdot \log_\alpha \frac{\sum_{e \in E_\Delta} \Delta(e)}{|E_\Delta|} \leq 4\alpha \cdot |E_\Delta| \leq 4am,$$

implying that $|E_\Delta| \geq g(Y_{\text{orig}})/(4\alpha)$. A valid mutation that chooses exactly one of the edges in $E_\Delta$ can be generated by the algorithm with probability $\Omega(|E_\Delta|/(e \cdot m))$, which results in a new solution $Y'$ with $g(Y') = g(Y_{\text{orig}}) - 1$. Thus the expected drift of $g$ can be bounded by

$$\frac{|E_\Delta|}{e \cdot m} \geq \frac{g(Y_{\text{orig}})}{e \cdot 4am}.$$ 

Case (2). $\sum_{e \in E_\Delta} \Delta(e) \geq \alpha \cdot |E_\Delta|$. By Lemma 3, we can get that

$$g(Y_{\text{orig}}) = \sum_{e \in E_\Delta} \beta(e) \leq \sum_{e \in E_\Delta} 2\alpha (\log_\alpha \Delta(e) + 1) \leq |E_\Delta| \cdot 2\alpha (\log_\alpha W_{\text{max}} + 1),$$

implying that $|E_\Delta| \geq \frac{g(Y_{\text{orig}})}{2\alpha (\log_\alpha W_{\text{max}} + 1)}$. Using the reasoning similar to that given for Case (1), we have that the expected drift of $g$ can be bounded by

$$\frac{|E_\Delta|}{e \cdot m} \geq \frac{g(Y_{\text{orig}})}{e \cdot 2am \cdot (\log_\alpha W_{\text{max}} + 1)}.$$

Now we consider the maximum value that $g(Y_{\text{orig}})$ can take,

$$g(Y_{\text{orig}}) = \sum_{e \in E_\Delta} \beta(e) \leq 2\alpha \cdot |E_\Delta| + 2\alpha \cdot \log_\alpha \left( \prod_{e \in E_\Delta} \Delta(e) \right)$$

(5)

$$\leq 2\alpha \cdot |E_\Delta| + 2\alpha \cdot |E_\Delta| \cdot \log_\alpha \frac{\sum_{e \in E_\Delta} \Delta(e)}{|E_\Delta|}$$

(6)

$$\leq 4\alpha \cdot |E_\Delta| \cdot \log_\alpha \frac{\sum_{e \in E_\Delta} \Delta(e)}{|E_\Delta|}$$

(7)

$$\leq 4\alpha \cdot |E_\Delta| \cdot \log_\alpha \frac{2D \cdot W_{\text{max}}}{|E_\Delta|}$$

(8)
where the factor $\log\alpha e$ is not greater than $e$ as $\alpha \in [2, W_{\text{max}}]$, and Inequality 9 can be derived by the observation that $f(x) = x \cdot \log_\alpha (2D \cdot W_{\text{max}}/x)$ ($x > 0$) gets its maximum value when $x = 2D \cdot W_{\text{max}}/e$.

Summarizing the analysis for Cases (1–2), we have that the expected drift of $g$ can be bounded by

\begin{equation}
\leq 4\alpha \cdot \frac{2D \cdot W_{\text{max}}}{e} \cdot \log_\alpha e
\end{equation}

and the maximum value of $g(Y_{\text{orig}})$ can be bounded by

\begin{equation}
\leq 8\alpha D \cdot W_{\text{max}}.
\end{equation}

The Multiplicative Drift Theorem [5] implies that the (1 + 1) EA takes expected runtime $O\left(\frac{am \log_\alpha W_{\text{max}} \cdot \log(\max\{am, \alpha D \cdot W_{\text{max}}\})}{e}\right)$ to find an MFDS of $G^*$ starting with $Y_{\text{orig}}$. Since we only consider the mutations selecting exactly one edge in the analysis for the (1 + 1) EA, the above conclusions also apply to the RLS. \qed

Consider an instance $\{G = (V, E, W), Y_{\text{orig}}, E^*\}$ of DWVC-E. By the analysis for DWVC-E$^+$ and DWVC-E$^-$, we have that $Y_{\text{orig}}$ is a feasible dual-solution of $G^* = (V, E^*, W)$, and the LP values of the edges in $E^+ \cup E_G(E^-)$ may have the room to be increased, where $E^+ = E^* \setminus E$ and $E^- = E \setminus E^*$. Since $|E^+ \cup E^-|$ is bounded by $D$, we can derive the following theorem for DWVC-E using the reasoning similar to that for Theorems 5 and 6.

**Theorem 7** The expected runtime of the (1 + 1) EA (or RLS) for DWVC-E is $O\left(\frac{am \log_\alpha W_{\text{max}} \cdot \log(\max\{am, \alpha D \cdot W_{\text{max}}\})}{e}\right)$.

### 4.2 Analysis for DWVC with Weight Modification

We start the subsection with the analysis of the (1 + 1) EA and RLS for the two special variants of DWVC-W, namely, DWVC-W$^+$ and DWVC-W$^-$. Denote by $V^+$ the set containing all the vertices $v$ with $W^*(v) > W(v)$, and by $V^-$ the set containing all the vertices $v$ with $W^*(v) < W(v)$. The variant DWVC-W$^+$ considers the case that $V^- = \emptyset$, and DWVC-W$^-$ considers the case that $V^+ = \emptyset$.

Consider an instance $\{G = (V, E, W), Y_{\text{orig}}, W^+, V^+\}$ of DWVC-W$^+$. Observe that $Y_{\text{orig}}$ is an obviously feasible dual-solution of $G^* = (V, E, W^*)$, and the LP values of the edges in $E_G(V^+)$ may have the room to be increased if $Y_{\text{orig}}$ is not an MFDS of $G^*$. The following lemma shows that the sum of the feasible LP value increments

\begin{align}
\sum_{v \in V} \frac{W^*(v) - W(v)}{e} &\leq 4\alpha \cdot \frac{2D \cdot W_{\text{max}}}{e} \cdot \log_\alpha e \\
&\leq 8\alpha D \cdot W_{\text{max}}.
\end{align}
on the edges in $E_{G^*}(V^+)$ can be upper bounded, as these edges are all incident to the vertices in $V^+$.

**Lemma 8** For any MFDS $Y^*$ obtained by the $(1 + 1) \text{EA}$ (or RLS) for the instance $\{G = (V, E, W), Y_{\text{orig}}, W^+, V^+\}$ of DWVC-W$^*$, 

$$\sum_{e \in E} (Y^*(e) - Y_{\text{orig}}(e)) \leq \sum_{v \in V^+} \left( W^+(v) - W(v) \right) \leq D \cdot W_{\text{max}}.$$

**Proof** Since $Y_{\text{orig}}$ is a feasible dual-solution of $G^*$, by Lemma 1, $Y^*(e) \geq Y_{\text{orig}}(e)$ for each edge $e \in E$. Let $E_{W^+}$ be the set containing all the edges $e \in E$ with $Y^*(e) > Y_{\text{orig}}(e)$. Then we have

$$\sum_{e \in E} (Y^*(e) - Y_{\text{orig}}(e)) = \sum_{e \in E_{W^+}} (Y^*(e) - Y_{\text{orig}}(e)). \quad (11)$$

Note that the LP values of the edges in $E \setminus E_{G^*}(V^+)$ cannot be increased, thus $E_{W^+} \subseteq E_{G^*}(V^+)$.

For each edge $e \in E_{W^+}$, let $\tau(e)$ be the endpoint of $e$ that is tight with respect to $Y_{\text{orig}}$ (if both endpoints of $e$ are tight, then arbitrarily choose one as $\tau(e)$). Observe that $\tau(e) \in V^+$ for each edge $e \in E_{W^+}$; otherwise, the LP value of the edge cannot be increased under the dual-LP constraint. Thus for any vertex $v \in V^+$, we have

$$\sum_{e \in E_{W^+} : \tau(e) = v} (Y^*(e) - Y_{\text{orig}}(e)) \leq W^+(v) - W(v). \quad (12)$$

Then summarizing Inequality (12) over all vertices in $V^+$, we can get

$$\sum_{e \in E_{W^+}} (Y^*(e) - Y_{\text{orig}}(e)) \leq \sum_{v \in V^+} \left( W^+(v) - W(v) \right) \leq D \cdot W_{\text{max}}. \quad (13)$$

Combining Equality (11) and Inequality (13) gives the claimed inequality. \qed

Using the reasoning similar to that for Theorem 6 and the upper bound given by Lemma 8, we can get the following theorem for DWVC-W$^*$.

**Theorem 9** The expected runtime of the $(1 + 1) \text{EA}$ (or RLS) for DWVC-W$^*$ is $O\left( a m \log a W_{\text{max}} \cdot \log(\max\{a m, a D \cdot W_{\text{max}}\}) \right)$. 

Given an instance $\{G = (V, E, W), Y_{\text{orig}}, W^-, V^-\}$ of DWVC-W$^-$, if $Y_{\text{orig}}$ is a feasible dual-solution of $G^* = (V, E, W^-)$, then $Y_{\text{orig}}$ is still an MFDS of $G^*$. Otherwise, we have to first decrease the LP values of the edges violating the dual-LP constraint with respect to the maintained dual-solution, as the general idea of the algorithms given in Sect. 3, to get the first feasible dual-solution as soon as possible. The remaining analysis to get an MFDS of $G^*$ based on the first feasible dual-solution is similar to that given for DWVC-E$^*$. 

\copyright Springer
Theorem 10 The expected runtime of the \((1 + 1)\) EA (or RLS) for DWVC-W is \(O(am \log_a W_{\max} \cdot \log(\max \{am, aD \cdot W_{\max}\}))\).

Proof We first analyze the expected runtime of the \((1 + 1)\) EA to obtain an MFDS \(Y^*\) of \(G^* = (V, E, W^-)\), starting with the MFDS \(Y_{\text{orig}}\) of \(G\). For the soundness and completeness of the proof, we assume that \(Y_{\text{orig}}\) is an infeasible dual-solution of \(G^*\). Then the whole process can be divided into Phase (I) and Phase (II). Phase (I) contains all steps of the algorithm until it finds the first feasible dual-solution \(Y^*_t\) of \(G^*\); Phase (II) follows Phase (I), which contains all steps of the algorithm until it obtains the MFDS \(Y^*\) of \(G^*\).

Phase (I) By Lemma 2, to get the first feasible dual-solution \(Y_t\) of \(G^*\), the \((1 + 1)\) EA only can decrease the LP values of the edges in \(E_{G^*}(Y_{\text{orig}})\), where \(E_{G^*}(Y_{\text{orig}}) \subseteq E_{G^*}(V^-)\). Thus \(Y_t(e) \leq Y_{\text{orig}}(e)\) for each edge \(e \in E_{G^*}(Y_{\text{orig}})\), and \(Y_t(e) = Y_{\text{orig}}(e)\) for each edge \(e \in E \setminus E_{G^*}(Y_{\text{orig}})\). Denote \(Y_{\text{orig}}(e) - Y_t(e)\) by \(\Delta(e)\) for each edge \(e \in E_{G^*}(Y_{\text{orig}})\), and denote by \(\beta(e)\) the number of valid mutations on \(e\) that the \((1 + 1)\) EA requires to decrease the LP value of \(e\) from \(Y_{\text{orig}}(e)\) to \(Y_t(e)\). Let \(E_A = \{e \in E_{G^*}(Y_{\text{orig}}) | \Delta(e) \neq 0\}\). Since each edge in \(E_A\) has an endpoint that is in \(V^-\), \(\sum_{e \in E_A} \Delta(e) \leq \sum_{e \in E_{\text{orig}}} Y_{\text{orig}}(e) \leq D \cdot W_{\max}\). Let the potential of the solution \(Y_{\text{orig}}(e)\) be

\[g(Y_{\text{orig}}(e)) = \sum_{e \in E_A} \beta(e).\]

Similar to the analysis given in Theorem 5, we have that \(g(Y_{\text{orig}})\) is the upper bound of the number of valid mutations on the edges in \(E_A\) that the algorithm requires to get \(Y_t\) starting from \(Y_{\text{orig}}\), and any mutation on \(Y_{\text{orig}}\) cannot increase its potential. The analysis for the expected drift of \(g\) is divided into two cases, based on the value of \(\sum_{e \in E_A} \Delta(e) / |E_A|\).

Case (1). \(\sum_{e \in E_A} \Delta(e) < \alpha \cdot |E_A|\). By Lemma 4, we have

\[\beta(e) \leq \lfloor \log_a ((\alpha - 1)\Delta(e) + 1) \rfloor \leq \lfloor \log_a (a\Delta(e)) \rfloor \leq \lfloor \log_a \Delta(e) + 1 \rfloor \leq \log_a \Delta(e) + 2\]

for each edge \(e \in E_A\). Thus

\[g(Y_{\text{orig}}) = \sum_{e \in E_A} \beta(e) \leq 2|E_A| + \log_a \left( \prod_{e \in E_A} \Delta(e) \right) \leq 2|E_A| + |E_A| \cdot \log_a \frac{\sum_{e \in E_A} \Delta(e)}{|E_A|} \leq 3|E_A|,\]

implying that \(|E_A| \geq g(Y_{\text{orig}})/3\), and the maximum value that \(g(Y_{\text{orig}})\) can take is \(3m\) (as \(|E_A| \leq m\)). A valid mutation that chooses exactly one of the edge in \(E_A\) can be generated by the algorithm with probability \(\Omega(|E_A| / (e \cdot m))\), which results in a new solution \(Y'\) with \(g(Y') = g(Y_{\text{orig}}) - 1\). Consequently, the expected drift of \(g\) can be bounded by

\[\sum_{e \in E_A} \Delta(e) / |E_A|\]
\[ \frac{|E_d|}{e \cdot m} \geq \frac{g(Y_{\text{orig}})}{3e \cdot m}. \]

Case (2). \( \sum_{e \in E_d} \Delta(e) \geq \alpha \cdot |E_d| \). By Lemma 4, we have that
\[ g(Y_{\text{orig}}) = \sum_{e \in E_d} \beta(e) \leq \sum_{e \in E_d} (\log_\alpha \Delta(e) + 2) \leq |E_d| \cdot (\log_\alpha W_{\text{max}} + 2), \]

implying that \( |E_d| \geq \frac{g(Y_{\text{orig}})}{\log_\alpha W_{\text{max}} + 2} \). Using the reasoning similar to that given for Case (1), we have that the expected drift of \( g \) can be bounded by
\[ \frac{|E_d|}{e \cdot m} \geq \frac{g(Y_{\text{orig}})}{e \cdot m \cdot (\log_\alpha W_{\text{max}} + 2)}. \]

Now we consider the maximum value that \( g(Y_{\text{orig}}) \) can take,
\[ g(Y_{\text{orig}}) = \sum_{e \in E_d} \beta(e) \leq 2|E_d| + \log_\alpha \left( \prod_{e \in E_d} \Delta(e) \right) \]
\[ \leq 2|E_d| + |E_d| \cdot \log_\alpha \frac{\sum_{e \in E_d} \Delta(e)}{|E_d|} \]
\[ \leq 3|E_d| \cdot \log_\alpha \frac{\sum_{e \in E_d} \Delta(e)}{|E_d|} \]
\[ \leq 3|E_d| \cdot \log_\alpha \frac{D \cdot W_{\text{max}}}{|E_d|} \]
\[ \leq 3 \cdot \frac{D \cdot W_{\text{max}}}{e} \cdot \log_\alpha e \]
\[ \leq 3D \cdot W_{\text{max}}, \]

where the factor \( \log_\alpha e \) is not greater than \( e \) as \( \alpha \in [2, W_{\text{max}}] \), and Inequality 18 can be derived by the observation that \( f(x) = x \cdot \log_\alpha (D \cdot W_{\text{max}}/x) \) \((x > 0)\) gets its maximum value when \( x = D \cdot W_{\text{max}}/e \).

Summarizing the analysis for Cases (1-2) and the fact that the value of \( \alpha \) is between 2 and \( W_{\text{max}} \), we have that the expected drift of \( g \) can be lower bounded by
\[ \frac{|E_d|}{e \cdot m} \geq \frac{g(Y_{\text{orig}})}{e \cdot m \cdot \max\{3, \log_\alpha W_{\text{max}} + 2\}} \geq \frac{g(Y_{\text{orig}})}{e \cdot m \cdot (\log_\alpha W_{\text{max}} + 2)}, \]

and the maximum value that \( g(Y_{\text{orig}}) \) can take is upper bounded by
The Multiplicative Drift Theorem [5] implies that the (1 + 1) EA takes expected runtime $O(m \log \alpha W_{\max} \cdot \log(\max\{m, D \cdot W_{\max}\}))$ to obtain the first feasible dual-solution $Y_t$ starting with $Y_{\text{orig}}$.

**Phase (II)** Obviously $Y_t$ may not be an MFDS of $G^* = (V, E, W^-)$. Thus we also need to consider the process of the (1 + 1) EA to get an MFDS of $G^*$ starting with $Y_t$. To simplify the analysis, we intend to transform Phase (II) as an execution of the (1 + 1) EA for an instance $\{G_t = (V, E, W_t), Y_t, W^-, V_t\}$ of DWVC-W$^+$. Thus in the following discussion, we first give the setting way of the weight function $W_t$ such that $Y_t$ is an MFDS of $G_t$, and $W_t(v) \leq W^-(v)$ for each vertex $v \in V$.

Let $V_\Delta$ contain all endpoints of the edges in $E_\Delta$. For each vertex $v \in V \setminus V_\Delta$, let $W'(v) = W(v)$, and for each vertex $v \in V_\Delta$, let

$$W'(v) = W(v) - \sum_{e \in E_\Delta \setminus \varnothing} \Delta(e).$$

As $Y_{\text{orig}}$ is an MFDS of $G$, $Y_t$ is an obvious MFDS of $G' = (V, E, W')$. Note that there may exist some vertex $v \in V$ with $W'(v) > W^-(v)$. Thus for each vertex $v \in V$, we let

$$W_t(v) = \min\{W'(v), W^-(v)\}. \tag{20}$$

Because of Equality 20 and the fact that $Y_t$ is a feasible dual-solution of both $G'$ and $G^*$, $Y_t$ is a feasible dual-solution of $G_t$. Furthermore, since $Y_t$ is an MFDS of $G'$, $Y_t$ is an MFDS of $G_t$. Now let $V_t$ contain all vertices $v \in V$ with $W_t(v) < W^-(v)$. Then the instance $\{G_t = (V, E, W_t), Y_t, W^-, V_t\}$ of DWVC-W$^+$ is completely constructed.

It is necessary to remark that for an edge $e$ in $E_\Delta$, the step size of $e$ may be larger than $Y_t^*(e) - Y_t(e)$ at the beginning of Phase (II), then Lemma 3 is invalid under the situation. Fortunately, the step size of $e$ is at most $\alpha \cdot (Y_{\text{orig}}(e) - Y_t(e)) = \alpha \cdot \Delta(e)$. Thus using the multiplicative drift analysis similar to that given above, we can get that the expected runtime of the (1 + 1) EA to decrease the step sizes of the edges in $E_\Delta$ to the feasible values is bounded by $O(m \log \alpha W_{\max} \cdot \log(\max\{m, D \cdot W_{\max}\}))$.

Now we assume that Lemma 3 is valid for each edge in $E_\Delta$. Similar to Lemma 8, we consider the upper bound on the sum of the feasible LP value increments on the edges with respect to $W^-$, where
\[
\sum_{v \in V_t} (W^+(v) - W_t(v)) \leq \sum_{v \in V_t} (W(v) - W_t(v)) \\
\leq \sum_{v \in V} ((W(v) - W^-(v)) + (W(v) - W'(v))) \\
\leq \sum_{v \in V} (W(v) - W^-(v)) + \sum_{e \in E_1 \cap \neg \emptyset} \Delta(e) \\
\leq \sum_{v \in V} (W(v) - W^-(v)) + 2 \sum_{e \in E_3} \Delta(e) \\
\leq D \cdot W_{\text{max}} + 2D \cdot W_{\text{max}} = 3D \cdot W_{\text{max}}.
\]

By Lemma 8 and the reasoning similar to that for Theorem 6, we have that the \((1 + 1)\) EA takes expected runtime \(O\left(\alpha \log_{\omega} W_{\text{max}} \cdot \log(\max\{\alpha, \alpha D \cdot W_{\text{max}}\})\right)\) to get an MFDS of \(G^*\) starting with \(Y_1\).

Summarizing the above discussion, the \((1 + 1)\) EA takes expected runtime \(O\left(\alpha \log_{\omega} W_{\text{max}} \cdot \log(\max\{\alpha, \alpha D \cdot W_{\text{max}}\})\right)\) to get an MFDS of \(G^*\). The above expected runtime also applies to the RLS.

**Theorem 11** The expected runtime of the \((1 + 1)\) EA (or RLS) for DWVC-W is \(O\left(\alpha \log_{\omega} W_{\text{max}} \cdot \log(\max\{\alpha, \alpha D \cdot W_{\text{max}}\})\right)\).

**Proof** Consider an instance \(\{G = (V, E, W), Y_{\text{orig}}, W^+, V^+, V^-\}\) of DWVC-W. By the discussion for DWVC-W+ and DWVC-W-, if \(Y_{\text{orig}}\) is a feasible dual-solution with respect to \(G^*\), then only the LP values of the edges in \(E_G(V^+\)} may have the room to be increased, and the sum of the increments can be upper bounded by \(D \cdot W_{\text{max}}\). If \(Y_{\text{orig}}\) is an infeasible dual-solution with respect to \(G^*\), then we have that the sum of the decrements on the LP values of the edges in \(E_G(V^-)\) and the sum of the increments on the LP values of the edges incident to the vertices in \(N_{G^*}(V^-) \cup V^- \cup V^+\) can be upper bounded by \(D \cdot W_{\text{max}}\) and \(3D \cdot W_{\text{max}}\), respectively. Using the reasoning similar to that for Theorems 6 and 10, we have the theorem for DWVC-W. \(\square\)

### 5 Runtime Analysis for the RLS with 1/5-th Rule and (1 + 1) EA with 1/5-th Rule

Given an MFDS \(Y^*\) that is obtained by the RLS with 1/5-th rule starting with a feasible dual-solution \(Y^+\) of \(G^*\), the following two lemmata consider the behavior of the algorithm on a specific edge \(e^*\) in \(G^*\) during the process from \(Y^+\) to \(Y^*\). Denote \(Y^+(e^*) - Y^+(e^*)\) by \(\Delta\).

**Lemma 12** If the step size of the edge \(e^*\) has an initial value \(\sigma_1 > 0\), then the RLS with 1/5-th rule takes expected runtime \(O\left(m(\log_{\omega} \sigma_1 + \log_{\omega} \Delta)\right)\) to find a feasible dual-solution \(Y^*\) such that \(Y^+(e^*) - Y^+(e^*) \geq \Delta/(\alpha + 1)\), during the process from \(Y^+\) to \(Y^*\).
Proof Since $Y^\dagger$ is a feasible dual-solution of $G^*$, any mutation that decreases the LP values of the edges would be rejected. Thus for any dual-solution $Y$ accepted by the algorithm during the process from $Y^\dagger$ to $Y^*$, there is $Y(e^*) \geq Y^\dagger(e^*)$. The following analysis divides the process from $Y^\sharp$ to $Y^\ddagger$ into two phases: Phase (I) and Phase (II).

As the initial value $\sigma_1$ of the step size of $e^*$ may be larger than $\Delta$, Phase (I) contains all steps of the algorithm until the step size of $e^*$ is not greater than $\Delta$, where $Y_1$ denotes the dual-solution maintained by the algorithm at that moment. We remark that for any dual-solution $Y$ (including $Y_1$) obtained during Phase (I), $Y(e^*) = Y^\dagger(e^*)$. Phase (II) follows Phase (I), and ends if the step size of $e^*$ is greater than $Y^*(e^*) - Y_2(e^*)$, where $Y_2$ denotes the dual-solution maintained by the algorithm at the moment. We will show that $Y_2$ satisfies the claimed condition given for $Y^\ddagger$. If $\sigma_1 \leq \Delta$, then we are already at Phase (II). For the soundness and completeness of the proof, we assume that $\sigma_1 > \Delta$ in the following discussion.

Let $\sigma_1 = \alpha^p$ and $\alpha^q \leq \Delta < \alpha^{q+1/4}$, where $p, q \in \{l/4 | l \in \mathbb{N}\}$. Now we analyze the expected runtime that Phase (I) takes to decrease the step size of $e^*$ from $\alpha^p$ to $\alpha^q$. Since $\sigma(e^*) > Y^*(e^*) - Y(e^*)$ always holds for any maintained dual-solution $Y$ during Phase (I), and the fact that $Y^\dagger$ is feasible, any mutation on $e^*$ cannot be accepted, and decreases the step size of $e^*$ by a factor $\alpha^{1/4}$. Thus Phase (I) needs $O(p - q) = O(p) = O(\log_\alpha \sigma_1)$ mutations on $e^*$ to decrease the step size to $\alpha^q$.

Now we assume that the step size of $e^*$ is decreased to $\alpha^q$, i.e., we are at Phase (II) now. If a mutation on $e^*$ increases its LP value, then the mutation would be accepted, and the exponent of the step size would be increased to $q + 1$; otherwise, the mutation would be rejected, and the exponent of the step size would be decreased to $q - 1/4$. The mutation on $e^*$ increases or decreases its LP value with the same probability 1/2, hence the value of the exponent increases by 1 or decreases by 1/4 with the same probability 1/2. Observe that the drift on the exponent is $(1 - 1/4)/2 = 3/8$. If the step size of $e^*$ is increased to over $\alpha^{[\log_\alpha \Delta]}$ (at most $\alpha^{[\log_\alpha \Delta + 1/4]}$), then Phase (II) obviously ends. In fact, the step size may not be increased to over $\alpha^{[\log_\alpha \Delta]}$ during Phase (II). Using the Additive Drift Theorem [10], the algorithm needs $O(\log_\alpha \Delta - q) = O(\log_\alpha \Delta)$ mutations on $e^*$ to increase the exponent to over $[\log_\alpha \Delta]$. Thus Phase (II) contains $O(\log_\alpha \Delta)$ mutations on $e^*$.

For the dual-solution $Y_2$ obtained by Phase (II), $\sigma(e^*) > Y^*(e^*) - Y_2(e^*)$. Since Phase (II) contains at least one mutation increasing the LP value of $e^*$ that is accepted, the gap $\Delta = Y^*(e^*) - Y^\dagger(e^*)$ is decreased by at least $\sigma(e^*)/\alpha$, and we have

$$Y^*(e^*) - Y_2(e^*) \leq \Delta - \sigma(e^*)/\alpha \leq \Delta - (Y^*(e^*) - Y_2(e^*))/\alpha.$$  

By the above inequality, it is easy to get that $Y_2(e^*) - Y^\dagger(e^*) \geq \Delta/(\alpha + 1)$.

Summarizing the above analysis, the algorithm takes expected runtime $O(m(\log_\alpha \sigma_1 + \log_\alpha \Delta))$ to get a feasible dual-solution satisfying the claimed condition. \hfill \square

Lemma 13 If the step size of the edge $e^*$ has an initial value not greater than $\Delta$, then the RLS with 1/5-th Rule takes expected runtime $O(m \log_\alpha \Delta \cdot \log \Delta)$ to increase the LP value of $e^*$ from $Y^\dagger(e^*)$ to $Y^*(e^*)$, during the process from $Y^\sharp$ to $Y^\ddagger$.  

© Springer
Proof Let \( Y \) be an arbitrary dual-solution obtained during the process from \( Y^\dagger \) to \( Y^* \). In the following discussion, we first analyze the expected runtime of the RLS with 1/5-th Rule to obtain a solution \( Y' \) starting with \( Y \) such that \( Y'(e^*) - Y(e^*) \geq (Y^*(e^*) - Y(e^*))/(\alpha + 1) \).

As the initial value of the step size of \( e^* \) is not greater than \( \Delta \), we have the observation that the maximum value of the step size of \( e^* \) during the process from \( Y^\dagger \) to \( Y^* \) is at most \( \alpha^{p+1} \), where \( p = \lceil \log_\alpha \Delta \rceil \). For the moment that the algorithm maintains the dual-solution \( Y \), the corresponding step size of \( e^* \) may be greater than \( Y^*(e^*) - Y(e^*) \), thus we have to consider Phase (I) (defined in the proof of Lemma 12). Since the step size of \( e^* \) is bounded by \( \alpha^{p+1} \), Phase (I) needs \( O(\log_\alpha \Delta) \) mutations on \( e^* \). Furthermore, since \( Y(e^*) \geq Y^\dagger(e^*) \), Phase (II) needs \( O(\log_\alpha (Y^*(e^*) - Y(e^*)) = O(\log_\alpha \Delta) \) mutations on \( e^* \). Consequently, the RLS with 1/5-th rule takes expected runtime \( O(m \log_\alpha \Delta) \) to get \( Y' \) starting with \( Y \).

Using the above conclusion, the Multiplicative Drift Theorem [5] implies the claimed runtime.

Now we analyze the expected runtime of the RLS with 1/5-th rule to make the edge \( e^* \) satisfy the dual-LP constraint, if it starts with an infeasible dual-solution with respect to which \( e^* \) violates the dual-LP constraint.

Lemma 14 Consider an infeasible dual-solution \( Y^\dagger \) of \( G^* \), with respect to which the edge \( e^* \) violates the dual-LP constraint. For the first feasible dual-solution \( Y^\dagger \) obtained by the RLS with 1/5-th rule starting with \( Y^\dagger \), the algorithm takes expected runtime \( O(m \log_\alpha (Y^\dagger(e^*) - Y^\dagger(e^*))) \) to decrease the LP value of \( e^* \) from \( Y^\dagger(e^*) \) to \( Y^\dagger(e^*) \).

Proof Assume that the step size of \( e^* \) has an initial value \( \alpha^q \), where \( q \geq 0 \). Since \( Y^\dagger \) is an infeasible dual-solution of \( G^* \), if a mutation on \( e^* \) decreases its LP value, then the mutation is accepted, and the exponent of the step size of \( e^* \) is increased by 1; otherwise, the mutation is rejected, and the exponent of the step size is decreased by \( 1/4 \). Observe that the mutation on \( e^* \) increases or decreases its LP value with the same probability 1/2. Thus we have that the drift on the exponent of the step size of \( e^* \) is \( (1 - 1/4)/2 = 3/8 \), during the process that decreases the LP value of \( e^* \) from \( Y^\dagger(e^*) \) to \( Y^\dagger(e^*) \).

If the step size of \( e^* \) is increased to not less than \( \alpha^{\lceil \log_\alpha (Y^\dagger(e^*) - Y^\dagger(e^*)) \rceil + 1} \) during the process that decreases the LP value of \( e^* \) from \( Y^\dagger(e^*) \) to \( Y^\dagger(e^*) \), then the LP value of \( e^* \) is decreased to 0. In fact, the step size may not be increased to over \( \alpha^{\lceil \log_\alpha (Y^\dagger(e^*) - Y^\dagger(e^*)) \rceil + 1} \) during the process, because \( \alpha^q \) may be greater than 0. Hence, the maximum value of the step size of \( e^* \) during the process is at most \( \alpha^{\lceil \log_\alpha (Y^\dagger(e^*) - Y^\dagger(e^*)) \rceil + 1} \). Using the Additive Drift Theorem [10] and the drift on the exponent of the step size of \( e^* \) obtained above, the process that decreases the LP value of \( e^* \) from \( Y^\dagger(e^*) \) to \( Y^\dagger(e^*) \) needs at most \( O(\log_\alpha (Y^\dagger(e^*) - Y^\dagger(e^*)) - q) \) \( = O(\log_\alpha (Y^\dagger(e^*) - Y^\dagger(e^*))) \) mutations on \( e^* \). That is, the process takes expected runtime \( O(m \log_\alpha (Y^\dagger(e^*) - Y^\dagger(e^*))) \). 

The following theorem can be derived based on the conclusions of Lemma 13.
Theorem 15 The expected runtime of the RLS with 1/5-th rule for DWVC-E⁺ is \(O(amD \log_m W_{\text{max}} \cdot \log W_{\text{max}}).\)

Proof We study the expected runtime of the RLS with 1/5-th Rule to obtain an MFDS \(Y^*\) of \(G^* = (V, E \cup E^+, W)\) starting with the MFDS \(Y_{\text{orig}}\) of \(G = (V, E, W)\). Observe that only the LP values of the edges in \(E^+\) may have the room to be increased. Moreover, Lemma 13 gives that for each edge \(e \in E^+\) with \(Y^*(e) > Y_{\text{orig}}(e)\), the algorithm takes expected runtime \(O(am \log_m W_{\text{max}} \cdot \log W_{\text{max}})\) to increase its LP value from \(Y_{\text{orig}}(e)\) to \(Y^*(e)\). Therefore, combining it with the fact that \(|E^+| = D\) directly gives the expected runtime \(O(amD \log_m W_{\text{max}} \cdot \log W_{\text{max}})\). \(\square\)

Theorem 16 The expected runtime of the RLS with 1/5-th rule for DWVC-E⁻ is \(O(am \log_m W_{\text{max}} \cdot \min\{m \log W_{\text{max}}, D \cdot W_{\text{max}}\})\).

Proof We study the expected runtime of the RLS with 1/5-th Rule to obtain an MFDS \(Y^*\) of \(G^* = (V, E \setminus E^-, W)\) starting with the given MFDS \(Y_{\text{orig}}\) of \(G = (V, E, W)\). Let \(E_d\) be the edges \(e \in E^* = E \setminus E^-\) with \(Y^*(e) > Y_{\text{orig}}(e)\), and let \(\Delta(e) = Y^*(e) - Y_{\text{orig}}(e)\) for each edge \(e \in E_d\). The reasoning given in Theorem 6 shows that \(\sum_{e \in E_d} \Delta(e) \leq D \cdot W_{\text{max}}\).

Lemma 13 gives that for each edge \(e \in E_d\), the algorithm takes expected runtime \(O(am \log_m W_{\text{max}} \cdot \log \Delta(e))\) to increase the LP value of \(e\) from \(Y_{\text{orig}}(e)\) to \(Y^*(e)\). Summing the expected runtime over all the edges in \(E_d\) gives the expected runtime \(O(am \log_m W_{\text{max}} \cdot \sum_{e \in E_d} \log \Delta(e))\). For the upper bound of \(\sum_{e \in E_d} \log \Delta(e)\), we have

\[
\sum_{e \in E_d} \log \Delta(e) = \log \left(\prod_{e \in E_d} \Delta(e)\right)
\]

\[
\leq |E_d| \cdot \log \frac{\sum_{e \in E_d} \Delta(e)}{|E_d|}
\]

\[
\leq |E_d| \cdot \log \frac{D \cdot W_{\text{max}}}{|E_d|}
\]

\[
\leq \frac{D \cdot W_{\text{max}}}{\log e} \cdot \log e.
\]

Inequality 24 can be obtained by the observation that \(f(x) = x \cdot \log(D \cdot W_{\text{max}}/x)\) \((x > 0)\) gets its maximum value when \(x = D \cdot W_{\text{max}}/e\). Thus the expected runtime of the RLS with 1/5-th rule to obtain an MFDS \(Y^*\) of \(G^*\) starting with \(Y_{\text{orig}}\) can be bounded by \(O(amD \log_m W_{\text{max}} \cdot W_{\text{max}})\).

Additionally, as \(|E_d| \leq m\), and \(\Delta(e) \leq W_{\text{max}}\) for each edge \(e \in E_d\), we have that the expected runtime of the algorithm can also be bounded by \(O(am^2 \log_m W_{\text{max}} \cdot \log W_{\text{max}})\). Therefore, combining the two expected runtime given above, we have the claimed result. \(\square\)
The following theorem can be derived using the conclusions obtained by Lemmata 13 and 14, and the reasoning similar to that given in Theorems 6, 10, and 16.

**Theorem 17** The expected runtime of the RLS with 1/5-th rule for DWVC-X, where \( X \in \{E, W^+, W^-, W\} \), is \( O(am \log_a W_{\text{max}} \cdot \min\{m \log W_{\text{max}}, D \cdot W_{\text{max}}\}) \).

In the following, we analyze the performance of the (1 + 1) EA with 1/5-th rule for DWVC. Firstly, we give a specific graph \( G_s \) (see Fig. 1a) that is the same as the graph considered in [21]. W.l.o.g., we assume that the maximum weight \( W_{\text{max}} \) that the vertices in \( G_s \) have is \( \alpha^n \). Then we show that in a special situation, the (1 + 1) EA with 1/5-th rule requires pseudo-polynomial runtime to obtain the unique MFDS \( Y^*_s \) of \( G_s \), where \( Y^*(e_i) = W_{\text{max}} \) and \( Y^*(e_i) = 1 \) for all \( 2 \leq i \leq m \). The Chernoff–Hoeffding Bound given below is used in the proof for the main result stated later.

**Chernoff–Hoeffding Bound** [20]. Let \( x_1, \ldots, x_n \) be independent random variables such that \( a_i \leq x_i \leq b_i \) for all \( 1 \leq i \leq n \). Denote \( X = \sum_{i=1}^{n} x_i \). Then for any \( \delta \geq 0 \), the following inequality holds.

\[
\text{Prob}(X \geq E[X] + \delta) \leq e^{-2\delta^2 / \sum_{i=1}^{n}(b_i-a_i)^2}.
\]

**Lemma 18** Consider a feasible dual-solution \( Y^\dagger \) of \( G_s \) with \( Y^\dagger(e_i) = 1 \) for all \( 1 \leq i \leq m \), and the step size of the edge \( e_1 \) with an initial value 1. The expected runtime of the (1 + 1) EA with 1/5-th rule to obtain the unique MFDS \( Y^* \) of \( G_s \) starting with \( Y^\dagger \) is lower bounded by \( 2^{m/2} \) \((0 < e \leq 1/2)\) with probability \( 1 - e^{-\Omega(m)} \).

**Proof** Let \( M \) be a mutation of the (1 + 1) EA with 1/5-th Rule that selects the edge \( e_1 \) (note that \( M \) may also select some edges in addition to \( e_1 \)). If \( M \) is accepted, then the exponent \( q \) of the step size \( \sigma(e_1) = \alpha^q \) of \( e_1 \) is increased by 1; otherwise, decreased by \( 1/4 \). Observe that \( M \) can be accepted only if \( M \) just selects the edge \( e_1 \) and increases its LP value. Thus the probability \( P_{\text{inc}} \) that the mutation \( M \) is accepted is \( \leq 1/2e \), and the probability \( P_{\text{dec}} \) that the mutation \( M \) is rejected is \( \geq 1 - 1/2e \). As the drift of the exponent \( q \) is \( 1 \cdot P_{\text{inc}} + (-1/4) \cdot P_{\text{dec}} \leq (5 - 2e)/8e < 0 \), \( q \) will gradually decrease to 0 if \( q > 0 \), i.e., the step size of \( e_1 \) will gradually decrease to 1 if it is greater than 1.

The step size of \( e_1 \) has an initial value 1, hence if it cannot increase to an enough large value during the whole process of \( 2^{m/2} \) steps with a high probability, then we can show that the LP value of \( e_1 \) after the \( 2^{m/2} \) steps cannot reach \( W_{\text{max}} = \alpha^n \) with a high probability. In the following discussion, we assume that the step size of \( e_1 \) is increased to \( \alpha \) at some point, i.e., \( q = 1 \). Then we use Chernoff–Hoeffding Bound to show that \( T_1 \) is upper bounded by \( m^e \) with probability \( 1 - e^{-\Omega(m^e)} \), where \( T_1 \) denotes the number of steps that the algorithm requires to decrease the step size of \( e_1 \) from \( \alpha \) to 1. Observe that \( x_i \) denotes the increment on the exponent \( q \) of the step size of \( e_1 \), which equals 1 or \(-1/4, a_i = -1/4, \) and \( b_i = 1 \) for all \( 1 \leq i \leq n \), where \( n = m^e \) (the notations \( x_i, b_i, \) and \( a_i \) follow the ones given in the definition of Chernoff–Hoeffding Bound). As the exponent \( q \) has values 1 and 0 before and after \( T_1 \) steps, respectively,
we have $X = \sum_{i=1}^{n} x_i = -1$. Furthermore, considering the equality $X = E[X] + \delta$, where $E[X] \leq \frac{2e}{8e} \cdot m^e$, Chernoff–Hoeffding Bound gives that the probability that $T_1 > m^e$ is upper bounded by

$$e^{-2\delta^2 / \sum_{i=1}^{n} (b_i - a_i)^2} = e^{-2[\frac{(2e^5}{8e} \cdot m^e - 1)]^2 / (\frac{25}{16} m^e)} = e^{-\Omega(m^e)}.$$ 

Meanwhile, we also have that the maximum value of the step size of $e_1$ during the $T_1$ steps is upper bounded by $\alpha^m$, with probability $1 - e^{-\Omega(m^e)}$.

Now we consider the whole process of $2m^{e/2}$ steps. A phase of the whole process is non-trivial if it starts with a point where the step size of $e_1$ is increased to $\alpha$, ends with a point where the step size of $e_1$ is decreased to 1 for the first time (i.e., the step sizes of $e_1$ at all internal points of the phase are greater than 1). Thus the whole process consists of $N_1$ non-trivial phases and $N_2$ steps where the step size of $e_1$ is 1 (both $N_1$ and $N_2$ are nonnegative integers). For a non-trivial phase $P$, by the analysis given above, the number of steps in $P$ is upper bounded by $m^e$ with probability $1 - e^{-\Omega(m^e)}$, and lower bounded by 5 (one step increases the step size to $\alpha$, and four steps decrease the step size to 1). Thus the number of non-trivial phases is upper bounded by $2m^{e/2}/5$. Combining the conclusion obtained above that the step size is increased to over $\alpha^m$ during each non-trivial phase with probability $e^{-\Omega(m^e)}$, for the $N_1$ non-trivial phases, we have that the step size of $e_1$ is increased to over $\alpha^m$ with probability

$$e^{-\Omega(m^e)} \cdot N_1 \leq e^{-\Omega(m^e)} \cdot 2m^{e/2}/5 = e^{-\Omega(m^e)}.$$ 

That is, during the whole process of $2m^{e/2}$ steps, the step size of $e_1$ is increased to over $\alpha^m$ with probability $e^{-\Omega(m^e)}$. Therefore, by the end of the whole process of $2m^{e/2}$ steps, the increment on the LP value of $e_1$ is upper bounded by $2m^{e/2} \cdot \alpha^m \leq \alpha^{2m^e}$ (as $\alpha \geq 2$) with probability $1 - e^{-\Omega(m^e)}$, where $\alpha^{2m^e}$ is less than $W_{\max} - 1 = \alpha^m - 1$ since $0 < e \leq 1/2$ and $m$ is sufficiently large. Therefore, with probability $1 - e^{-\Omega(m^e)}$, the $(1 + 1)$ EA with 1/5-th rule cannot find the unique MFDS of $G^*$ within runtime $2m^{e/2}$.

---

**Fig. 1** a The special graph $G_s$ contains $m$ edges, each of which is independent (i.e., each edge constitutes a connected component of $G_s$). Except the two endpoints of edge $e_1$ in $G_s$ that have weight $W_{\max}$, all the other vertices have weight 1. b The graph $G_{s}'$ is a variant of $G_s$ with an additional vertex and an additional edge $e'_1$. 

© Springer
There are two reasons for the pseudo-polynomial runtime of the (1 + 1) EA with 1/5-th rule for DWVC-E+: (1) the small probability of a mutation to be accepted by the algorithm; (2) the “radical” strategy that decreases the step sizes of all the edges selected by the mutation if it is rejected. Under the combined impact of the two factors, the step size of $e_1$ cannot be increased to an enough large value to overcome the exponential large weight $W_{\text{max}}$. An obvious workaround is incorporating the “conservative” strategy (adopted by Algorithm 1) into the (1 + 1) EA with 1/5-th rule, which only decreases the step sizes of the edges that satisfy a strict condition. Then the probability of a mutation that decreases the step size of $e_1$ would be smaller.

Another possible workaround is considering the $1/i$-th rule, where $i > 5$, to slow down the decreasing speed of the step size of $e_1$. Both workarounds aim to make the expected drift of the step size of $e_1$ be positive.

Considering the instance $\{G_s \setminus \{e_1\}, Y_{\text{orig}}^+, E^+ = \{e_1\}\}$ of DWVC-E+, where $Y_{\text{orig}}(e_i) = 1$ for all $2 \leq i \leq m$, we can get that Theorem 19 holds for DWVC-E+ by Lemma 18. Similarly, considering the instance $\{G_s', Y_{\text{orig}}^-, E^- = \{e'_1\}\}$ of DWVC-E−, where $Y_{\text{orig}}(e_i) = 1$ for all $1 \leq i \leq m$ and $Y_{\text{orig}}(e'_1) = W_{\text{max}} - 1$ (graph $G_s'$ is given in Fig. 1b), we can get that Theorem 19 holds for DWVC-E− by Lemma 18. Considering that the weights of the two endpoints of $e_1$ in $G_s$ are increased from 1 to $W_{\text{max}}$, and $Y_{\text{orig}}(e_i) = 1$ for all $1 \leq i \leq m$, we can get that Theorem 19 holds for DWVC-W+ by Lemma 18. Considering that the weight of the endpoint of $e'_1$ that is not shared with $e_1$ in $G_s'$ is decreased from $W_{\text{max}}$ to 1, and $Y_{\text{orig}}(e'_1) = W_{\text{max}} - 1$ and $Y_{\text{orig}}(e_i) = 1$ for all $1 \leq i \leq m$, we can get that Theorem 19 holds for DWVC-W− by Lemma 18. Combining the conclusions for DWVC-E+, DWVC-E−, DWVC-W+, and DWVC-W−, we have that Theorem 19 holds for DWVC-E and DWVC-W.

**Theorem 19** If the maximum weight $W_{\text{max}}$ of the vertices in the considered weighted graph is $\alpha''m$, then the expected runtime of the (1 + 1) EA with 1/5-th rule for DWVC-X, where $X \in \{E^+, E^-, E, W^+, W^-, W\}$, is lower bounded by $2^{m/2} \cdot (0 < \epsilon \leq 1/2).$

**6 Conclusion**

In the paper, we contributed to the theoretical understanding of evolutionary computing for the dynamic weighted vertex cover problem, generalizing the results obtained by Pourhassan et al. [22] for the dynamic vertex cover problem. Two graph-editing operations were studied for the dynamic changes on the given weighted graph, which lead to two versions: dynamic weighted vertex cover problem with edge modification and dynamic weighted vertex cover problem with weight modification, and two special variants for each version.

We first introduced two algorithms (1 + 1) EA and RLS with the step size adaption strategy, and analyzed their performances for the two versions (including their four special variants) separately. Our analysis shows that the qualities of the solutions for these studied dynamic changes can be maintained efficiently. As mentioned in
Sect. 1, Pourhassan et al. [21] studied the weighted vertex cover problem using the dual form of the LP formulation, and showed that their \((1 + 1)\) EA with step size adaption strategy cannot get a 2-approximate solution in polynomial expected runtime with a high probability. It is easy to find that our \((1 + 1)\) EA can be extended to solve the weighted vertex cover problem efficiently (i.e., construct a 2-approximate solution), of which each instance \(G' = (V', E', W')\) can be transformed to an instance of DWVC-E\(^+\) with \(E = \emptyset\) and \(E^+ = E'\). There are two main differences between their \((1 + 1)\) EA and our \((1 + 1)\) EA, causing the big performance gap: (1). for the mutation \(M\) of their \((1 + 1)\) EA, the adjustment directions of the LP values of the edges selected by \(M\) are random, i.e., there may exist two edges selected by \(M\) whose LP values are increased and decreased respectively; for the mutation \(M\) of our \((1 + 1)\) EA, the LP values of the edges selected by \(M\) are either all increased or all decreased, and the adjustment direction depends on the feasibility of the maintained solution; (2). for the mutation \(M\) that is rejected by their \((1 + 1)\) EA, the step sizes of all the edges selected by \(M\) are decreased; for the mutation \(M\) that is rejected by our \((1 + 1)\) EA, only the step sizes of the edges satisfying a specific condition can be decreased.

To eliminate the artificial influences on the behaviors of the two algorithms mentioned above, we also incorporated the 1/5-th (success) rule to control the increasing/decreasing rate of the step size, and presented two algorithms named \((1 + 1)\) EA with 1/5-th rule and RLS with 1/5-th rule. The RLS with 1/5-th rule was shown to be able to maintain the qualities of the solutions efficiently as well. However, for the \((1 + 1)\) EA with 1/5-th rule, its performance was shown to be not satisfying. More specifically, its expected runtime is lower bounded by a pseudo-polynomial with a high probability, to maintain the qualities of the solutions if the maximum weight that the vertices have is exponential with respect to the number of the edges in the graph. The result matches that given by Pourhassan et al. [21], and indicates that the 1/5-th rule cannot overcome the negative impact caused by the standard mutation operator, when considering the special instances. However, for the \(1/i\)-th rule with a sufficiently large value of \(i\), it seems to be a promising way to overcome such an impact.

This is the first work that incorporates the 1/5-th rule with the step size adaption strategy, to solve a dynamic combinatorial optimization problem. We will leave it for future research to extend these insights to the analysis for more (dynamic) combinatorial optimization problems.

Acknowledgements The work is supported by the National Natural Science Foundation of China under Grants 61802441, 61672536, 61836016, and the Australian Research Council (ARC) through Grant DP160102401.

References

1. Auger, A., Doerr, B.: Theory of Randomized Search Heuristics: Foundations and Recent Developments. World Scientific Publishing Co. Inc., Singapore (2011)
2. Bar-Yehuda, R., Even, S.: A linear-time approximation algorithm for the weighted vertex cover problem. J. Algorithms 2(2), 198–203 (1981)
3. Beyer, H.G., Schwefel, H.P.: Evolution strategies—a comprehensive introduction. Nat. Comput. 1(1), 3–52 (2002)
4. Doerr, B., Doerr, C.: Optimal parameter choices through self-adjustment: applying the 1/5-th rule in discrete settings. In: Proceedings of the 2015 Genetic and Evolutionary Computation Conference, pp. 1335–1342. ACM (2015)
5. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. Algorithmica 64(4), 673–697 (2012)
6. Du, D.Z., Ko, K.I., Hu, X.: Design and Analysis of Approximation Algorithms, vol. 62. Springer, Berlin (2011)
7. Friedrich, T., He, J., Hebbinghaus, N., Neumann, F., Witt, C.: Approximating covering problems by randomized search heuristics using multi-objective models. Evol. Comput. 18(4), 617–633 (2010)
8. Friedrich, T., Neumann, F.: What’s hot in evolutionary computation. In: AAAI, pp. 5064–5066 (2017)
9. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. 11(1), 1–18 (2003)
10. He, J., Yao, X.: A study of drift analysis for estimating computation time of evolutionary algorithms. Nat. Comput. 3, 21–35 (2004)
11. Hochbaum, D.S.: Efficient bounds for the stable set, vertex cover and set packing problems. Discrete Appl. Math. 6(3), 243–254 (1983)
12. Jansen, T.: Analyzing Evolutionary Algorithms—The Computer Science Perspective. Natural Computing Series. Springer, Berlin (2013)
13. Jansen, T., Oliveto, P.S., Zarges, C.: Approximating vertex cover using edge-based representations. In: Proceedings of the 12th ACM/SIGEVO Workshop on Foundations of genetic algorithms (FOGA XII), pp. 87–96. ACM (2013)
14. Khot, S.: On the power of unique 2-prover 1-round games. In: Proceedings of the 34th Annual ACM Symposium on Theory of computing, pp. 767–775. ACM (2002)
15. Khot, S., Regev, O.: Vertex cover might be hard to approximate to within 2-ε. J. Comput. Syst. Sci. 74(3), 335–349 (2008)
16. Kötzing, T., Lissovoi, A., Witt, C.: (1+1) EA on generalized dynamic onemax. In: Proceedings of the 13th ACM/SIGEVO Conference on Foundations of Genetic Algorithms (FOGA XIII), pp. 40–51. ACM (2015)
17. Kratsch, S., Neumann, F.: Fixed-parameter evolutionary algorithms and the vertex cover problem. Algorithmica 65(4), 754–771 (2013)
18. Neumann, F., Witt, C.: Bioinspired computation in combinatorial optimization: algorithms and their computational complexity, 1st edn. Springer, New York (2010)
19. Neumann, F., Witt, C.: On the runtime of randomized local search and simple evolutionary algorithms for dynamic makespan scheduling. In: Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI), pp. 3742–3748 (2015)
20. Phillips, J.M.: Chernoff-hoeffding inequality and applications (2012). arXiv preprint arXiv:1209.6396
21. Pourhassan, M., Friedrich, T., Neumann, F.: On the use of the dual formulation for minimum weighted vertex cover in evolutionary algorithms. In: Proceedings of the 14th ACM/SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA XIV), pp. 37–44. ACM (2017)
22. Pourhassan, M., Gao, W., Neumann, F.: Maintaining 2-approximations for the dynamic vertex cover problem using evolutionary algorithms. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 903–910. ACM (2015)
23. Pourhassan, M., Shi, F., Neumann, F.: Parameterized analysis of multi-objective evolutionary algorithms and the weighted vertex cover problem. In: Proceedings of the 14th International Conference on Parallel Problem Solving from Nature (PPSN XIV), pp. 729–739. Springer (2016)
24. Roostapour, V., Neumann, A., Neumann, F.: On the performance of baseline evolutionary algorithms on the dynamic knapsack problem. In: the 15th International Conference on Parallel Problem Solving from Nature (PPSN XV), pp. 158–169. Springer (2018)
25. Shi, F., Neumann, F., Wang, J.: Runtime analysis of randomized search heuristics for the dynamic weighted vertex cover problem. In: Proceedings of the 2018 Genetic and Evolutionary Computation Conference, pp. 1515–1522. ACM (2018)
26. Shi, F., Schirneck, M., Friedrich, T., Kötzing, T., Neumann, F.: Reoptimization time analysis of evolutionary algorithms on linear functions under dynamic uniform constraints. Algorithmica 81(2), 828–857 (2019)
27. Vazirani, V.V.: Approximation Algorithms. Springer, Berlin (2013)

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