# Time Complexity Analysis of RLS and (1+1) EA for the Edge Coloring Problem

Jakob Bossek
bossek@wi.uni-muenster.de
Dept. of Information Systems
University of Münster, Münster, Germany

Dirk Sudholt
d.sudholt@sheffield.ac.uk
Department of Computer Science
University of Sheffield, Sheffield, United Kingdom

## ABSTRACT

The edge coloring problem asks for an assignment of colors to edges of a graph such that no two incident edges share the same color and the number of colors is minimized. It is known that all graphs with maximum degree $\Delta$ can be colored with $\Delta$ or $\Delta + 1$ colors, but it is $\mathcal{NP}$-hard to determine whether $\Delta$ colors are sufficient.

We present the first runtime analysis of evolutionary algorithms (EAs) for the edge coloring problem. Simple EAs such as RLS and (1+1) EA efficiently find $(2\Delta - 1)$-colorings on arbitrary graphs and optimal colorings for even and odd cycles, paths, star graphs and arbitrary trees. A partial analysis for toroids also suggests efficient runtimes in bipartite graphs with many cycles. Experiments support these findings and investigate additional graph classes such as hypercubes, complete graphs and complete bipartite graphs. Theoretical and experimental results suggest that simple EAs find optimal colorings for all these graph classes in expected time $O(\Delta\ell^2 m \log m)$, where $m$ is the number of edges and $\ell$ is the length of the longest simple path in the graph.

## CCS CONCEPTS

• **Theory of computation → Theory of randomized search heuristics**;

## KEYWORDS

Edge coloring problem, runtime analysis

**ACM Reference Format:**
Jakob Bossek and Dirk Sudholt. 2019. Time Complexity Analysis of RLS and (1+1) EA for the Edge Coloring Problem. In *Proceedings of Foundations of Genetic Algorithms XV (FOGA '19).* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3299904.3340311

## 1 INTRODUCTION

Evolutionary algorithms (EAs) are general purpose, bio-inspired methods that have proven to perform extraordinarily well on a wide range of optimization problems [7]. In the last decades theoretical analysis of EAs and related methods, especially the time complexity analysis, gained a lot of attention and came up with a manifold of analysis methods and results [2, 15, 22]. While in the beginning simple toy functions like ONEMAX were studied predominantly, research soon turned its focus to well-known combinatorial optimization problems. So far many results are available for, e. g., minimum spanning trees [21], maximum matchings [13], shortest paths [24], Eulerian cycles [20], scheduling [28] or vertex coloring [11, 25, 26] to mention a few.

We contribute to the fundamental understanding of working principles of evolutionary algorithms by considering the edge coloring problem. Given a simple graph with $n$ vertices and $m$ edges, the goal is to assign colors to the edges such that no two incident edges share the same color, called a proper coloring, and the size of the used color palette is minimal. Edge coloring has various applications in (job shop) scheduling [27] or the assignment of frequencies in fiber optic networks [10].

One can easily see that each simple graph can be properly colored with $2\Delta - 1$ colors in time $O(\Delta m)$, where $\Delta$ is the maximum node degree in the graph. An astonishing theorem by Vizing states that any simple graph can be colored either by $\Delta$ (class 1) or $\Delta + 1$ (class 2) colors.[1] Holyer [14] proved that edge coloring is $\mathcal{NP}$-hard on general graphs and hence all known exact algorithmic endeavours require exponential time. However, Misra & Gries [19] provided a constructive proof of Vizing's theorem. The resulting algorithm finds an coloring with at most $\Delta + 1$ colors in time $O(nm)$ and makes use of a sophisticated procedure called a fan rotation. Oftentimes, restrictions to specific graph classes lead to more efficient algorithms since one can leverage structural properties. For $\Delta$-edge-colorable bipartite graphs, algorithms with running time $O(m \log m)$ by Alon [1], $O(\Delta m)$ by Schrijver and $O(m \log \Delta)$ by Cole, Ost and Schirra [8] have been proposed.

In distributed computing many machines operate collaboratively in order to solve a problem. In the famous LOCAL communication model a network is expressed as a graph $G = (V, E)$ with maximum degree $\Delta$ where adjacent nodes perform local computations and exchange information directly in discrete rounds via message passing (see, e. g., [18]). Here, the running time is expressed by means of the (expected) number of rounds until a given problem is solved and one usually aims for poly-logarithmic running times. Graph coloring problems (vertex and edge coloring) have a long tradition in distributed computing and have been studied extensively for decades mainly due to their symmetry breaking properties and the fact that it is easy to check solutions locally [3]. That is, to verify that a $(2\Delta - 1)$-edge-coloring is proper for a certain edge one needs to check the local neighborhood only. Simple, yet efficient local

*FOGA '19, August 27–29, 2019, Potsdam, Germany*

ACM ISBN 978-1-4503-6254-2/19/08…$15.00
https://doi.org/10.1145/3299904.3340311

[1] Examples for class 1 graphs are even cycles, bipartite graphs in general and complete graphs with an even number of nodes. In contrast, e. g., odd cycles and complete graphs with an odd number of nodes belong to class 2.

randomized distributed algorithms for $(2\Delta - 1)$-coloring exist for over 30 years [3]. Beating this natural barrier in the deterministic setting was a major open problem for decades. Just recently, considerable progress was made [4, 6, 12], e. g., by Ghaffari et al. [12] who proposed a deterministic distributed algorithm that calculates a $(1 + o(1))\Delta$-coloring in poly-logarithmic time in the local model as long as $\Delta = \omega(\log n)$.

Graph coloring has been subject of studies in the context of randomized search heuristics [11, 25, 26]. This includes studies of a simple Ising model, a model of ferromagnetism where one seeks to *maximize* the number of edges where both end points have the same color, as it is equivalent to the vertex coloring problem in case of bipartite graphs. For the Ising model/vertex coloring, Fischer and Wegener [11] showed that on cycle graphs RLS and (1+1) EA find a proper 2-coloring in expected time $O(n^3)$ and $O(n^2)$ if crossover is used. Sudholt [25] considered the class of complete binary trees and showed that (1+1) EA needs exponential expected time, but a simple Genetic Algorithm with fitness sharing and crossover locates a global optimum in expected cubic time. Sudholt and Zarges [26] studied the running time of an iterated local search (ILS) algorithm with different mutation operators based on color eliminations and Kempe chains (as in the algorithm of Misra and Gries). These operators recolor large connected parts of the graph. They showed that ILS with color eliminations efficiently computes 2-colorings in bipartite graphs while ILS with Kempe chains needs exponential time with overwhelming probability. Recently, Bossek et al. [5] studied vertex coloring in a dynamic setting where edges are added to a properly colored graph over time. Their results show that re-optimization can be much faster than optimization from scratch in certain situations. In contrast, adding edges in an unfavorable order may lead to even worse asymptotic running times than in the static setting.

In contrast to vertex coloring, the edge coloring problem has not been considered by the EA theory community, despite being a well-known $\mathcal{NP}$-hard problem with important applications. We address this problem here by providing rigorous runtime analyses of RLS and (1+1) EA for selected graph classes. We show that these algorithms are able to find proper edge colorings efficiently for a range of graph classes. Our main results are gathered in Table 1.

This work is structured as follows. After formulating the foundations in Section 2 we given some general bounds in Section 3. We prove that a proper $(2\Delta - 1)$-coloring can be found in expected time $O(\Delta m \log m)$ on arbitrary simple graphs with maximum degree $\Delta$, consider the expected time to find colorings with few conflicts, and formulate general lower bounds for general graphs. Next, we shift our focus to optimal colorings. In Section 4 we provide upper bounds for simple graph classes, e. g., cycles, paths and star graphs. In Section 5 we show that on every tree the expected time to find a proper coloring with $\Delta$ colors is bounded from above by $O(\Delta \ell^2 m \log m)$ in expectation where $\ell$ is the length of the longest path in the tree. In Section 6 we discuss the analysis of toroid graphs as an example of a graph class with multiple cycles. Since the analysis turns out to be surprisingly challenging, we only present a rigorous lower bound for a worst-case initialization and discuss the challenges involved in rigorously proving upper bounds. Section 7 joins theory and practice by conducting a series of experiments to (1) empirically back up our results, such as a conjectured $O(m^3)$ bound for toroids, and (2) to check assumptions on other, more general graph classes (e. g., hypercubes and complete graphs). Section 8 completes our first excursion into edge coloring with concluding remarks and pointers to promising future research directions.

The appendix contains tools for the analysis of fair random walks used in the main part; our presentation of these largely known results may be of independent interest.

## 2 PRELIMINARIES

Let $G = (V, E)$ be a simple undirected graph with $n = |V|$ and $m = |E|$. For an edge $e$ we denote by $\deg(e)$ the number of edges incident to $e$ and by $N(e)$ the set of edges incident to $e$. Note that for every graph with minimum degree $\delta$ and maximum degree $\Delta$ and every edge $e$ we have

$$2\delta - 2 \leq \deg(e) \leq 2\Delta - 2.$$

By $\ell := \ell(G)$ we denote the length of the longest simple path in $G$ (that is, a path that does not loop back on itself).

We call a function $c : E \to \{1, \ldots, k\}$ an *edge coloring/coloring* of $G$. An edge coloring $c$ is termed *proper* if no two incident edges share the same color, i. e., $\forall\, e_1, e_2 \in E$: $e_1 \cap e_2 \neq \emptyset \Rightarrow c(e_1) \neq c(e_2)$. A graph is *$k$-colorable* if there is a proper edge coloring with $k$ colors. The smallest number $k$, such that $G$ is $k$-colorable, is the so-called *chromatic index* and denoted $\chi'(G)$ or just $\chi'$ in the following.

We call an edge pair $(e_1, e_2)$ with $e_1 \neq e_2$ a *conflict* if the edges are incident and have the same color. Likewise, we call an edge $e$ a *conflict edge* if there is at least one edge in $N(e)$ that has the same color assigned. We shall often refer to the unique vertex shared between $e_1$ and $e_2$ as the *common vertex* of the conflict. A color $i$ is termed *free* for $e$ if no incident edge is colored with $i$.

### 2.1 Algorithms

In this work we consider the size of the color palette to be fixed to a parameter $k \geq \chi'$. The search space is thus given by $S = \{1, \ldots, k\}^m$ and the fitness function used is to minimize the number of conflicts, that is, the number of edge pairs that are conflicting. For example, if there are 4 edges $e_1, e_2, e_3, e_4$ sharing a common vertex and all colored identically, they contribute $\binom{4}{2} = 6$ conflicts to the fitness.

Clearly, a solution $x \in S$ with zero fitness is a proper $k$-coloring. We are interested in the expected number of function evaluations required until simple randomized search heuristics locate a proper coloring for the first time. The algorithms under consideration are randomized local search (RLS, see Algorithm 1) and (1+1) EA (see Algorithm 2). Both algorithms maintain a single incumbent solution $x$ which is initialized uniformly at random. In each iteration the incumbent is subject to mutation and the mutant $y$ replaces $x$ if it has no more conflicts. The only difference is in the mutation operator. While RLS recolors a single edge in each iteration with probability 1 (called a *local move*), (1+1) EA recolors each edge with probability $1/m$ independently. It thus has the ability to perform multiple local moves simultaneously.

Unless stated otherwise, RLS and (1+1) EA start with a coloring generated uniformly at random. Most of the upcoming positive results will hold for arbitrary initial colorings.

**Table 1: Overview of presented results. Here $m$ is the number of edges, $\Delta$ the maximum degree, $\delta$ the minimum degree, $\ell$ the length of the longest simple path, $k$ is the size of the color palette and $\chi'$ is the chromatic index. The lower bound for (bipartite) toroids holds for a worst-case initial coloring with two remaining conflicts. We conjecture an upper bound of $O(m^3)$ for RLS on bipartite toroids from any initialization.**

| | Graph class | Colors $k$ | RLS | (1+1) EA |
|---|---|---|---|---|
| **Coloring with $k = \chi'$** | Cycle (even) | $\Delta$ | $O(m^3)$ [Thm 14] | $O(m^3)$ [Thm 14] |
| | Cycle (odd) | $\Delta + 1$ | $O(m \log m)$ [Thm 15] | $O(m \log m)$ [Thm 15] |
| | Path | $\Delta$ | $O(m^3)$ [Thm 16] | $O(m^3)$ [Thm 16] |
| | Star | $\Delta$ | $O(m^2)$ [Thm 17] | $O(m^2)$ [Thm 17] |
| | General tree | $\Delta$ | $O(\Delta \ell^2 m \log m)$ [Thm 18] | – |
| | Toroid | $\Delta$ | $\Omega(m^3)$ [Thm 24] | – |
| **Coloring with $k > \chi'$** | Graph with restricted edge neighborhood | $\Delta + \tau + 1$ | $O(\Delta m \log m)$ [Thm 6] | $O(\Delta m \log m)$ [Thm 6] |
| | Every graph | $2\Delta - 1$ | $O(\Delta m \log m)$ [Cor 7] | $O(\Delta m \log m)$ [Cor 7] |
| **General lower bounds** | Every connected graph | $\geq \Delta$ | $\Omega(m \log(m/k))$ [Thm 11] | $\Omega(m \log(m/k))$ [Thm 11] |
| | Every connected graph | $\delta + O(1)$ | $\Omega(km)$ [Thm 12] | $\Omega(km)$ [Thm 12] |

**Algorithm 1** RLS

```
Generate $x \in \{1, \dots, k\}^m$ uniformly at random.
while optimum not found do
    Generate $y$ by choosing an index $i \in \{1, \dots, m\}$ uniformly
    at random, choosing a new value $y_i \in \{1, \dots, k\}$ uniformly
    at random and setting $y_j = x_j$ for all $j \neq i$.
    If $y$ has no more conflicts than $x$, let $x := y$.
```

**Algorithm 2** (1+1) EA

```
Generate $x \in \{1, \dots, k\}^m$ uniformly at random.
while optimum not found do
    Generate $y$ by deciding to mutate each edge $x_i$ with prob-
    ability $1/m$: if yes, choose a new value $y_i \in \{1, \dots, k\}$
    uniformly at random.
    If $y$ has no more conflicts than $x$, let $x := y$.
```

It should be noted that for reasons of clarity and consistency – and because it seems to be more natural for *edge* coloring – all our runtime bounds are stated in terms of the number of edges, $m$, as opposed to the number of vertices.

## 2.2 On the Effect of Local Moves

To lay the foundations for the upcoming analyses, we explain the effect of local moves, before considering fitness-improving and fitness-neutral local moves (that is, local moves not altering the fitness) in more detail.

Consider a local move at time $t$ changing the color of an edge $e = \{u, v\}$ from $i$ to $j \neq i$. This move can only affect the status of edges in $N(e) \cup \{e\}$.

Let $e_1, e_2, \dots, e_k$ be all $i$-colored edges in $N(e)$ (if any). Then the recolor operation will resolve all conflicts $(e, e_1), (e, e_2), \dots, (e, e_\ell)$. However, if $e'_1, e'_2, \dots, e'_r$ are all $j$-colored edges in $N(e)$ (if any) then the move will create conflicts $(e, e'_1), (e, e'_2), \dots, (e, e'_r)$.

It will be useful to regard conflicts as particles that can move through the graph. For example, if one previously conflicting edge pair $(e, e')$ becomes non-conflicting but another edge pair $(e, e'')$ now becomes conflicting, we say that the conflict has moved from $(e, e')$ to $(e, e'')$. If a local move at $e$ reduces the number of conflicts by $s$, we select $s$ conflicts involving $e$ uniformly at random and declare these to be resolved. The remaining conflicts (if any) are then declared to have moved.

This random selection is used to break symmetries and to ensure that every conflict has a fair chance to be removed in a fitness-improving local move. For instance, if $N(e)$ contains two $i$-colored edges $e_1, e_2$ and one $j$-colored edge $e'_1$ then either the conflict $(e, e_1)$ moves to $(e, e'_1)$ and the conflict $(e, e_2)$ is declared resolved, or the conflict $(e, e_2)$ moves to $(e, e'_1)$ and the conflict $(e, e_1)$ is declared resolved. These decisions are made with equal probability.

## 2.3 On Possible Improvements

We first collect some statements that allow us to identify possible improvements.

Recall that a color $i$ is called a free color for an edge $e$ if color $i$ does not appear in the neighborhood of $e$. For every conflict $(e_1, e_2)$, if either edge $e_1$ or $e_2$ has a free color, there is a local move that resolves the conflict.

The following lemma lower-bounds the number of free colors, or colors that only lead to one conflict.

Lemma 1. *For every edge $e$ the following holds. Let $k_{\text{free}}$ be the number of free colors at $e$ and $k_{\text{one}}$ the number of colors that only* *create one conflict among its incident edges, then*

$$2k_{\text{free}} + k_{\text{one}} \geq 2k - \deg(e).$$

*In particular, if there is no free color for e then e has at least* $2k - \deg(e)$ *colors leading to one conflict only.*

Proof. Note that $k_{\text{one}}$ colors account for $k_{\text{one}}$ edges incident to $e$. All $k_{\text{free}}$ free colors do not contribute any edges, but all remaining $k - k_{\text{free}} - k_{\text{one}}$ colors contribute at least 2 edges. Since there are only $\deg(e)$ edges, we have

$$(k - k_{\text{free}} - k_{\text{one}}) \cdot 2 + k_{\text{one}} \cdot 1 + k_{\text{one}} \cdot 0 \leq \deg(e)$$

which is equivalent to $2k_{\text{free}} + k_{\text{one}} \geq 2k - \deg(e)$. □

By Lemma 1 every edge $e$ involved in a conflict either has a free color or it has one other color that leads to one conflict. We refer to the latter color as an *alternative color.*

For edges that are part of many conflicts, there is a larger probability of reducing the number of conflicts.

Lemma 2. *For every edge e that is part of at least* 3 *conflicts there are at least* $k - \Delta + \lceil(\Delta - 1)/3\rceil$ *other colors for e that lead to at most 2 conflicts.*

Proof. There are at most $2\Delta - 2$ edges incident to $e$. There can be at most $\lfloor(2\Delta - 2)/3\rfloor$ colors that also lead to 3 (or more) conflicts. Thus there must be $k - 1 - \lfloor(2\Delta - 2)/3\rfloor \geq k - \Delta + (\Delta - 1)/3$ other colors that have at most 2 conflicts. Since the number of colors is an integer, it is at least $k - \Delta + \lceil(\Delta - 1)/3\rceil$ as claimed. □

Conflicts can be resolved in case two or more conflicts of the same color are incident.

Lemma 3. *For every graph G with maximum degree* $\Delta$*, for every conflict* $(e_1, e_2)$ *the following holds. If the conflict is incident to another conflict* $(e_3, e_4)$ *of the same color, with probability at least* $1/(2km)$ *the conflict* $(e_1, e_2)$ *is resolved in the next step.*

Proof. Note that edges $e_1, e_2, e_3, e_4$ may not be mutually different, however we know that $e_1 \neq e_2$, $e_3 \neq e_4$ and $s := \left|\bigcup_{i=1}^{4}\{e_i\}\right| \geq 3$ as we are dealing with two different conflicts. We consider the following cases:

(1) The union of the two conflicts contains a path of length at least 3.
(2) The two conflicts share a common center vertex.

Note that these are the only cases since the absence of a path of length at least 3 implies that all edges must have one vertex in common.

In the first case, the middle edge of that path can be recolored with another color that only creates at most 1 conflict. By Lemma 1 this reduces the number of conflicts. Since we may create a new conflict, the conflict $(e_1, e_2)$ may either be declared as resolved, or declared to have moved to the new conflicting edges. Since there are at least 2 conflicts affected by the recolor operation, one of these will be chosen uniformly at random to be declared to have moved. So with probability at least 1/2, conflict $(e_1, e_2)$ will be declared as resolved.

In the second case, each of the $s$ edges in $\bigcup_{i=1}^{4}\{e_i\}$ is involved in at least $s - 1 \geq 2$ conflicts. According to Lemma 1, recoloring an edge in $\{e_1, e_2\}$ with a free or an alternative color will make at least 2 edge pairs non-conflicting (including $(e_1, e_2)$) while making at most one non-conflicting edge pair conflicting. As above, the probability that $(e_1, e_2)$ will be declared resolved is at least 1/2. □

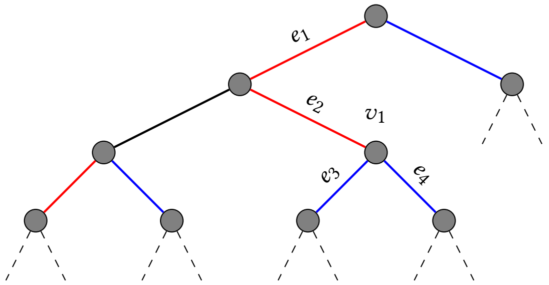

**Figure 1: Example of a blocked conflict $(e_1, e_2)$. Here, it is blocked by another conflict $(e_3, e_4)$ and cannot move further down.**

## 2.4 Fitness-Neutral Operations

We also describe and characterize some fitness-neutral operations.

*Definition 4.* Let $(e_1, e_2)$ be a conflict at time $t$ and let $(e_1', e_2') \neq (e_1, e_2)$ be the same conflict at time $t+1$. We say that the conflict *was rotated* at time $t$ if the common vertex has not changed: $e_1 \cap e_2 = e_1' \cap e_2'$. Otherwise, that is, if the common vertex has changed to a neighbouring vertex, we say that the conflict *has moved.*

The following lemma establishes that conflicts can move in the graph unless they are blocked by other conflicts.

Lemma 5. *Consider a conflict* $(e_1, e_2)$*. Let* $v_1$ *be the end point of* $e_1$ *not shared with* $e_2$*. If there is no conflict that has* $v_1$ *as shared vertex then* $e_1$ *has a free color or an alternative color that, when applied, leads to the conflict being moved, with* $v_1$ *as the new shared vertex.*

*The same statement also holds with the roles of* $e_1$ *and* $e_2$ *swapped.*

Proof. W. l. o. g. $e_1$ has color 1. Call $v$ the unique joint vertex in $e_1 \cap e_2$. We pessimistically assume that $e_1$ has no free color as otherwise the statement is trivial. By Lemma 1 (and excluding the color $c(e_1)$ itself) $e_1$ has $s := 2k - \deg(e) - 1$ alternative colors, w. l. o. g. color 2. We prove the statement by contraposition. Assume that all these alternative colors lead to the conflict being rotated. Then for all alternative colors $i \in A(e_1)$, there is exactly one $i$-colored edge at $v$ and there is no $i$-colored edge at $v_1$, as otherwise $i$ would not be an alternative color for $e_1$.

This means that the number of colors used at $v_1$ is at most $k - (2k - \deg(e) - 1) = \deg(e) - k + 1 = (\deg(v_1) + \deg(v) - 2) - k + 1 \leq \deg(v_1) - 1$. By the pigeon-hole principle, there must be at least one color that appears at least twice at $v_1$. This completes the contraposition. Hence if there is no conflict with $v_1$ as shared vertex, there must be an alternative color for $e_1$ that moves the conflict along the edge $e_1$, with $v_1$ as new shared vertex. □

The requirements of Lemma 5 are necessary. Assume there is another conflict $(e_3, e_4)$, $e_3 \neq e_1$ and $e_4 \neq e_1$, with $e_1$'s only alternative color $c(e_3) \neq c(e_1)$ (assuming $e_1$ only has one alternative color) and $v_1$ as its common vertex (see Figure 1). Then recoloring $e_1$ with its alternative color $c(e_3)$ yields two conflicting edge pairs, $(e_1, e_3)$ and

$(e_1, e_4)$. Unless there are further $c(e_1)$-colored conflicts involving $e_1$, this move leads to a decrease in fitness and will be rejected by RLS. We say that then the conflict $(e_1, e_2)$ is *blocked* by the conflict $(e_3, e_4)$.

# 3 GENERAL BOUNDS

## 3.1 $(2\Delta - 1)$-Coloring Arbitrary Graphs

We first concentrate on rather generous color palettes such that local recoloring can always reduce the number of conflicts. Our first result is on arbitrary graphs with restricted edge neighborhoods.

Theorem 6. *On every graph $G = (V, E)$ with maximum degree $\Delta$ and $\max_{e \in E} \deg(e) = \Delta + \tau$ for $0 \le \tau \le \Delta - 2$, for every initial coloring, RLS and (1+1) EA find a proper coloring with $k = \Delta + \tau + 1$ colors in expected time $O(\Delta m \log m)$.*

Proof. First note that given a color palette of size $k = \Delta + \tau + 1$ for each conflict edge there is always at least one free color. Now let $X_t \in \mathbb{N}$ denote the number of conflicts and $X_t(e)$ the number of conflicts edge $e$ is part of at time $t$. Clearly, $X_t \le \binom{m}{2} = x_{\max}$ and – since every edge is counted twice – $\sum_e X_t(e) = 2X_t$. With probability at least $1/(ekm)$, (1+1) EA resolves all $X_t(e)$ conflicts the edge $e$ is involved in. This lower bound does hold for RLS, too. As a consequence we get

$$E(X_{t+1} \mid X_t) \le X_t - \frac{\sum_e X_t(e)}{ekm} = X_t - \frac{2X_t}{ekm} = X_t \left(1 - \frac{2}{ekm}\right).$$

This implies an expected drift of

$$E(X_t - X_{t+1} \mid X_t) \ge X_t \left(\frac{2}{ekm}\right).$$

At last we apply the multiplicative drift theorem [9] and obtain a runtime bound of

$$\frac{ekm}{2} \ln(1 + O(m^2)) = O(\Delta m \log m).$$

The final equality is due to $k = \Theta(\Delta)$. □

Note that every simple graph admits a proper coloring with $2\Delta - 1$ colors, since $\deg(e) \le 2\Delta - 2$ for all edges. Hence, setting $\tau = \Delta - 2$ in Theorem 6 we obtain the following result.

Corollary 7. *On every graph with maximum degree $\Delta$ and for every initial coloring, RLS and (1+1) EA find a coloring with $k = 2\Delta - 1$ colors in expected time $O(\Delta m \log m)$.*

## 3.2 Reducing the Number of Conflicts

The number of conflicts (that is, the number of conflicting edge pairs) can be as large as $\binom{m}{2} = \Theta(m^2)$ for a star graph or a complete graph where all edges have the same color. We show that, for every number of colors $k \ge \Delta$ and every initial coloring, the number of conflicts quickly decreases to at most $m$.

Theorem 8. *For every graph $G$ with $m$ edges and maximum degree $\Delta$ and every initial coloring, the expected time until RLS or (1+1) EA with $k \ge \Delta$ colors have found a solution with at most $m$ conflicts is $O(m \log m)$.*

Proof. Let $X_t$ again denote the number of conflicts at time $t$ and $X_t(e)$ denote the number of conflicts edge $e$ is part of at time $t$.

For edges $e$ with $X_t(e) > 2$, Lemma 2 states that there are at least $k - \Delta + \lceil(\Delta - 1)/3\rceil$ alternative colors which lead to at most two conflicts. The probability of executing a local move at $e$ is at least $1/(em)$ for both RLS and (1+1) EA and the probability to recolor the sampled edge with one of these colors is at least

$$\begin{aligned}\frac{k - \Delta + \lceil(\Delta - 1)/3\rceil}{k} &= 1 - \left(\frac{\Delta - \lceil(\Delta - 1)/3\rceil}{k}\right)\\ &\ge 1 - \left(\frac{\Delta - \lceil(\Delta - 1)/3\rceil}{\Delta}\right) \ge \frac{1}{4}.\end{aligned}$$

Here, the last inequality stems from the observation that the term in braces is maximal for $\Delta = 4$. Thus, if $X_t(e) > 2$ then with probability at least $1/(4em)$ we get $X_t(e) - X_{t+1}(e) \ge X_t(e) - 2$. The same statement holds trivially for $X_t(e) \le 2$ as a local move at $e$ cannot increase the number of conflicts. As long as $X_t \ge m + 1$, the overall expected drift is thus

$$\begin{aligned}E(X_t - X_{t+1} \mid X_t) \ge \sum_e \frac{1}{4em} \cdot (X_t(e) - 2)\\ = \frac{1}{4em}\left(\sum_e X_t(e) - 2m\right)\\ = \frac{1}{4em}(2X_t - 2m).\end{aligned}$$

The variable drift theorem [16] (Theorem 3 in [17]) then yields an upper bound of

$$\begin{aligned}&2em + \int_{m+1}^{m^2} \frac{4em}{2x - 2m}\,\mathrm{d}x\\ &= 2em + 4em \int_{m+1}^{m^2} \frac{1}{2x - 2m}\,\mathrm{d}x\\ &= 2em + 4em \left[\frac{\ln(2x - 2m)}{2}\right]_{m+1}^{m^2}\\ &= 2em + 4em \left(\frac{\ln(2m^2 - 2m)}{2} - \frac{\ln 2}{2}\right) \le 2em + 4em \ln m.\end{aligned}$$ □

We remark without proof that within expected time $O(\Delta m \log m)$, both RLS and (1+1) EA find a solution with at most $m/2$ conflicts, for every graph with $m$ edges. This can be shown by waiting for free or alternative colors to be applied, similarly to the proof of Theorem 6.

## 3.3 General Lower Bounds

To complement our upper bounds and to establish a baseline for good performance, we now turn to proving lower bounds for RLS and (1+1) EA. We show two lower bounds that apply to arbitrary connected graphs. As a first step, we show that the initial coloring has $\Theta(m/k)$ conflicts with high probability.

Lemma 9. *For every connected graph with $m$ edges, if an edge coloring is chosen uniformly at random with colors $\{1, \ldots, k\}$ then there are at least $m/(4k)$ conflicts on mutually disjoint edges, with probability $1 - e^{-\Omega(m)}$.*

To show Lemma 9, we first show the following combinatorial result.

Lemma 10. *Every connected graph with $m$ edges admits a sequence of mutually disjoint edges $e_1, \dots e_{2\lfloor m/2 \rfloor}$ such that for all $i$ with $1 \leq i \leq \lfloor m/2 \rfloor$, $e_{2i-1}$ and $e_{2i}$ are incident edges.*

Proof. We prove the claim by induction over $m$. The claim is trivial for $m \leq 1$. Assume that the claim holds for $m-2$ and consider any connected graph $G$ with $m \geq 2$ edges.

Consider a spanning tree $T$ of $G$ rooted at an arbitrary but fixed vertex and denote by $\deg_T(v)$ the degree of a vertex $v$ in $T$. We consider leaves in $T$ that have a maximum graph distance from the root in the subgraph induced by $T$ and call these *deepest leaves.*

If there is a deepest leaf $u$ that is incident to at least two edges not belonging to $T$, we remove two such edges. All vertices remain connected via $T$ and hence we can decompose the remaining graph of $m-2$ edges inductively into $\lfloor (m-2)/2 \rfloor = \lfloor m/2 \rfloor - 1$ further edge pairs.

Otherwise, if there is a deepest leaf $u$, with a parent that we call $v$, that is incident to exactly one edge $e$ not belonging to $T$, we remove $e$ and $\{u, v\}$ and the graph remains connected via $T \setminus \{u, v\}$ since $u$ is a leaf in $T$. The remaining graph can then be decomposed inductively.

Otherwise all deepest leaves in $T$ are also leaves in $G$. Fix a deepest leaf $u$ with a parent that we call $v$. If $u$ has a sibling $u'$ in the tree then $u'$ must be another deepest leaf as $u$ was chosen to have a maximum graph distance to the root. Then the edges $\{u, v\}$ and $\{u', v\}$ are incident. Removing these edges leaves a connected graph with $m-2$ edges, which can be decomposed inductively.

If $u$ does not have a sibling in $T$ then, since there are at least 2 edges in the graph, $v$ must have a parent in the tree that we call $w$. We note that $\{u, v\}$ and $\{v, w\}$ are incident, and removing them from the graph leaves a connected graph as there are no further edges at $u$ nor $v$. Removing $\{u, v\}$ and $\{v, w\}$ and decomposing the remaining graph inductively as above completes the proof. □

Proof of Lemma 9. By Lemma 10 there is a sequence of mutually disjoint edges $e_1, \dots, e_{2\lfloor m/2 \rfloor}$ such that edges $e_{2i-1}$ and $e_{2i}$ are incident, for all $1 \leq i \leq \lfloor m/2 \rfloor$. For each such edge pair the probability that the two edges will be conflicting after a random initialization is $1/k$. These events are independent for all edge pairs, hence we can apply Chernoff bounds. This yields that, with probability $1 - e^{-\Omega(m)}$, at least $m/(4k)$ edge pairs $e_{2i-1}, e_{2i}$ are conflicting after initialization. □

The following lower bound follows now from Lemma 9 and standard coupon collector arguments.

Theorem 11. *The expected time for RLS or (1+1) EA to find a proper $k$-coloring on any connected graph $G$, for any value of $k \leq m$, is $\Omega(m \log(m/k))$. This is $\Omega(m \log m)$ if $k = O(m^{1-\Omega(1)})$; this is the case, for example, for all regular graphs or graphs where $\delta = \Omega(\Delta)$.*

Proof. By Lemma 9, with probability $1 - e^{-\Omega(m)}$, the initial coloring has at least $m/(4k)$ conflicts on mutually disjoint edges. Assume that this happens and fix a conflicting pair. The conflict can only be resolved if one of the two edges is being picked during mutation. The probability for this event is at most $2/m$ for both RLS and (1+1) EA.

The probability that a fixed conflicting pair is not resolved within $t := (m/2 - 1) \ln(m/(4k))$ mutations is at least

$$\left(1 - \frac{2}{m}\right)^t \geq e^{-\ln(m/(4k))} = \frac{4k}{m}.$$

The probability that there is a conflict out of the $m/(4k)$ conflicts that is not resolved after time $t$ is at least

$$1 - \left(1 - \frac{4k}{m}\right)^{m/(4k)} \geq 1 - \frac{1}{e}.$$

This means that the expected optimization time is at least $(1 - 1/e - e^{-\Omega(m)}) \cdot t = \Omega(m \log(m/k))$. □

Theorem 11 shows that the upper bound from Corollary 7 is asymptotically tight if $\Delta = O(1)$ as then $\log(m/k) = \Theta(\log m)$.

We also give a lower bound that includes a factor of $k$ (but no $\log m$ factor).

Theorem 12. *The expected time of RLS and (1+1) EA to find a proper $k$-coloring on any connected graph with minimum degree $\delta \geq 2$ and $\delta \leq k \leq m$ is at least $\Omega(km/(k - \delta + 1))$. This is $\Omega(km)$ if $k = \delta + O(1)$, for example when the graph is $\Delta$-regular and $k \in \{\Delta, \Delta + 1\}$.*

Proof. By Lemma 9, the probability of initializing with an optimal solution is $e^{-\Omega(m)}$.

The best case situation for finding a proper coloring is attained when there is just one conflict $(e_1, e_2)$, or two conflicts that share an edge $e_1$ and form a path of 3 edges. This is because if there are two conflicts with disjoint edge pairs, or multiple conflicts that have the same vertex as common vertex, multiple specific edges need to be recolored to find the optimum in one step. This is impossible for RLS and has probability $O(1/m^2)$ for (1+1) EA.

To find a proper coloring from a coloring with just one conflict, it is necessary to recolor $e_1$ or $e_2$. For any such edge $e$, since there are no other conflicts, at least $\delta - 1$ colors are taken, hence the number of free colors is at most $k - \delta + 1$. The probability to recolor one of the two involved edges and to pick a free color is at most $2(k - \delta + 1)/(km)$ and the expected waiting time for this event is at least $km/(2(k - \delta + 1))$.

In the case of two conflicts with a shared edge $e_1$, $e_1$ must be recolored with one of $k - \delta + 1$ free colors, which has probability at most $(k - \delta + 1)/(km)$. The expected waiting time in this case is at least $km/(k - \delta + 1)$. □

Together, we obtain the following result for $\Delta$-regular graphs.

Corollary 13. *The expected time for RLS and (1+1) EA to find a proper coloring on any $\Delta$-regular connected graph, with $k \leq \Delta + O(1)$, is $\Omega(\Delta m + m \log m)$.*

## 4 RUNTIME BOUNDS FOR SIMPLE GRAPH CLASSES

We now consider the performance of RLS and (1+1) EA on a range of simple graph classes. We start with cycle graphs, that is, graphs that consist of a single cycle visiting all vertices.

Theorem 14. *For every initial coloring, the expected time for RLS and (1+1) EA to find a proper 2-coloring on every cycle graph $C_{2n}$ with an even number of nodes is $O(m^3)$.*

Proof. We first review the notion of a *line-graph* $L(G)$ of an arbitrary graph $G$. The line-graph is a graph with one node for each edge of $G$ and an edge between nodes if and only if the corresponding edges in $G$ are incident. It is easy to see that an optimal edge-coloring of $G$ corresponds to an optimal vertex-coloring of $L(G)$ and vice versa. Cycle graphs have the appealing property that its line-graph is again a cycle graph with $n$ nodes and $m = n$ edges. Since $\chi'(G) = \chi(L(G))$ we can focus on vertex-coloring of $L(G)$ with 2 colors. An equivalent problem is to minimize the number of monochromatic blocks which was studied by Fischer and Wegener [11] in context of the Ising model on rings (cycle graphs). The authors prove an upper bound of $O(n^3)$ function evaluations for RLS and (1+1) EA respectively. This result directly implies a runtime of $O(m^3)$ for edge-coloring of $C_{2n}$ with 2 colors. The key idea of [11] is to consider connected monochromatic blocks and the length of the shortest block in particular. They estimate the number of so-called relevant steps, i. e., steps that either decrease the number of monochromatic blocks or the length of the shortest block by $O(n^2)$. The key argument is that the algorithms need to overcome plateaus of length at most $n/2$. Here, random walk arguments yield the quadratic bound. Since $n$ such steps are sufficient we end up with a runtime bound of $O(n^3) = O(m^3)$. □

Note that cycle graphs with an odd number of edges do not admit a proper 2-coloring and hence at least three colors are needed. The additional color makes the problem much easier, because there is always a free color for a conflicting edge.

Theorem 15. *For every initial coloring, the expected time of RLS and (1+1) EA to find a proper 3-coloring on a cycle graph $C_{2n+1}$ with an odd number of nodes is $O(m \log m)$.*

Proof. Note that in $C_{2n+1}$ we have $2\Delta - 1 = 3$ and hence the theorem follows directly from Corollary 7. □

We also note for completeness that paths can be colored in the same way as even cycles, with almost identical proofs.

Theorem 16. *For every initial coloring, the expected time of RLS and (1+1) EA to find a proper 2-coloring on a path with m edges is $O(m^3)$.*

Proof. Follows the same arguments as the proof of Theorem 14. □

Now we consider star graphs, defined as a graph with a vertex in the center of the graph, to which all edges are incident. This implies $\Delta = m$.

Theorem 17. *The expected time of RLS and (1+1) EA to find a proper coloring with $k = \Delta = m$ colors on a star graph with m edges is bounded by $O(m^2)$.*

Proof. Consider the number of conflicts $X_t$ at time $t \in \mathbb{N}_0$ and denote by $X_t(i)$ the number of edges colored with color $i \in \{1, \ldots, m\}$ at time $t$. Note that $X_t(i) \geq 1$ implies that there are $(X_t(i) - 1)$ edges which shall be colored differently with free colors. Note further that

$$X_t = \sum_{i=1}^{m} \binom{X_t(i)}{2} = \frac{1}{2} \sum_{i=1}^{m} X_t(i) \cdot (X_t(i) - 1).$$

Call the total number of free colors $s$. With the considerations from above we can conclude $s = \sum_{i=1}^{m} \max\{0, X_t(i) - 1\}$. The max-function ensures that colors that are not used so far do not have a negative contribution to $s$.

Note that in both $X_t$ and $s$ all values $X_t(i) \leq 1$ lead to a contribution of 0, hence we can focus on values $X_t(i) \geq 2$. Using $X_t(i) \leq 2(X_t(i) - 1)$ for $X_t(i) \geq 2$,

$$\begin{aligned} X_t &= \frac{1}{2} \sum_{i\colon X_t(i) \geq 2} X_t(i) \cdot (X_t(i) - 1) \\ &\leq \sum_{i\colon X_t(i) \geq 2} (X_t(i) - 1)^2 \\ &\leq \Big( \sum_{i\colon X_t(i) \geq 2} (X_t(i) - 1) \Big)^2 \\ &= \Big( \sum_{i\colon X_t(i) \geq 2} \max\{0, X_t(i) - 1\} \Big)^2 = s^2, \end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality. We conclude that $s \geq \sqrt{X_t}$.

If $X_t > 1$ we can improve by selecting a single edge and recoloring this edge with a free color. This happens with probability at least $s/(ekm) \geq s/(em^2)$ for both RLS and (1+1) EA. Hence, the overall expected drift is

$$\begin{aligned} E(X_t - X_{t+1} \mid X_t) &\geq \frac{1}{2} \sum_{i=1}^{m} X_t(i) \cdot (X_t(i) - 1) \cdot \Big( \frac{s}{em^2} \Big) \\ &= X_t \cdot \Big( \frac{s}{em^2} \Big) \\ &\geq \frac{X_t^{3/2}}{em^2}. \end{aligned}$$

With $x_{\min} = 1 \leq X_t \leq \binom{m}{2} < m^2$ the variable drift theorem yields an upper bound of

$$\begin{aligned} & em^2 + \int_1^{m^2} \left(em^2\right) x^{-3/2} \, \mathrm{d}x \\ &= em^2 + em^2 \int_1^{m^2} x^{-3/2} \, \mathrm{d}x \\ &= em^2 + em^2 \left[ -\frac{2}{\sqrt{x}} \right]_1^{m^2} = em^2 + em^2 \left( 2 - \frac{2}{m} \right) \\ &\leq 3em^2 - 2em = O(m^2). \end{aligned}$$

□

## 5 A BOUND FOR TREES

We now show that RLS can efficiently edge-color arbitrary trees with $\Delta$ colors. We focus on RLS instead of (1+1) EA as the analysis becomes more involved. Even on simple graphs such as cycles, Fischer and Wegener's work shows that the analysis of (1+1) EA becomes way more complicated than that of RLS [11] and it is not clear whether (1+1) EA has any advantage over RLS (we shall revisit this question experimentally, in Section 7).

Theorem 18. *On every tree G with $\ell := \ell(G)$, RLS with $k = \Delta$ finds a proper $\Delta$-coloring in expected time $O(\Delta \ell^2 m \log m)$.*

Proof. Let $h$ be the height of the tree, i. e., the length of the longest simple path from the root to any leaf. Note that $h \leq \ell \leq 2h$, hence we only need to show an upper bound of $O(\Delta h^2 m \log n)$. For a vertex $v$ denote by $d(v)$ the depth of $v$, that is, the length of the unique simple path from $v$ to the root.

We identify the initial conflicts with tags $1, 2, 3, \dots$ that move with the conflicts. Once a conflict is resolved, the tag disappears; until this happens, the tag is called *active*. We denote by $c(i)$ the color of the conflict tagged $i$. Define $\varphi_t(i) := h - d(v_t(i))$, where $v_t(i)$ denotes the common vertex of the conflict tagged $i$ at time $t$. If the tag has disappeared from the graph, we define $\varphi_t(i) := 0$. Note that, while the tag is active, $1 \leq \varphi_t(i) \leq h$ as the common vertex of any conflict cannot be a leaf, hence $0 \leq d(v_t(i)) \leq h - 1$.

By Lemma 5, a conflict can move up or down in the tree as long as it is not blocked by another conflict (see Figure 1 for an example of a blocked conflict). While there is no blocking conflict, there is only at most one recolor operation that would move a conflict closer to the root, thus increasing $\varphi_t$, while there is at least one recolor operation that would move it away from the root, thus decreasing $\varphi_t$. While $\varphi_t(i) = 1$ the conflict has reached a leaf and can be resolved by recoloring the edge incident to the leaf. However, blocking conflicts complicate the situation as they can eliminate moves that decrease $\varphi_t$. On the other hand, the blocking conflict has an advantage as it does not have any moves that can increase $\varphi_t$. We address this by considering the following model that reflects how conflicts move through the tree.

Consider a conflict tagged $i$ and an edge $e$ that connects levels $d(v_t(i))$ and $d(v_t(i) + 1)$. Assume that $e$ is incident to both edges of another conflict tagged $j$ on levels $d(v_t(i) + 1)$ and $d(v_t(i) + 2)$. If a recolor operation picks edge $e$ and color $c(j)$ then we *swap tags i and j*. This is done regardless of whether the recolor operation is accepted or not. The idea behind this swap is that while a conflict may be blocked by another conflict, tags can roam more freely. We will show in the following that the $\varphi_t$-values of tags are stochastically dominated by a fair random walk.

Lemma 19. *For every tag i we have*

$$\Pr(\varphi_{t+1}(i) = \varphi_t(i) + 1) \leq \frac{1}{km}.$$

Proof. A tag can only move up under the following conditions. If the two edges of conflict $i$ are on the same level, the only way the tag can move up is if it is swapped with a tag higher up in the tree. This requires a specific recolor operation that occurs with probability $1/(km)$.

If the two edges of the conflict are on different levels, tag $i$ cannot be swapped "upwards", but a recolor operation can move the conflict up. Let $e_1$ be the upper edge of the conflict and $e_2$ be the unique edge incident to $e_1$ on the level above. For a recolor operation to move the conflict up, $e_1$ must be recolored with color $c(e_2)$. This operation has probability $1/(km)$. □

Lemma 20. *For every active tag i we have*

$$\Pr(\varphi_{t+1}(i) \leq \varphi_t(i) - 1) \geq \frac{1}{km}.$$

Proof. Consider an edge $e = \{v_1, v_t(i)\}$ of the conflict where $d(v_1) = d(v_t(i)) + 1$.

First assume that $e$ has a free color. Note that this is implied by $\varphi_t(i) = 1$ as then $e$ is incident to a leaf; since two edges at $v$ are colored $c(i)$ and $\deg(e) = \deg(v) \leq \Delta$ there must be a free color by the pigeon-hole principle. Choosing a free color would remove the tag, yielding $\varphi_{t+1}(i) = 0 \leq \varphi_t(i) - 1$. The probability of applying a free color to $e$ is at least $1/(km)$.

Now assume that $e$ has no free color, which implies $\varphi_t(i) \geq 2$. By Lemma 5, if there is no other conflict that has $v_1$ as shared vertex, there must be a free color or an alternative color that moves the conflict further down in the tree, leading to $\varphi_{t+1}(i) = \varphi_t(i) - 1$. The probability for this event is at least $1/(km)$.

Finally, we assume that $e$ has no free color but there is a conflict tagged $j$ with $v_1$ as shared vertex. We consider two sub-cases. First assume that $e$ is incident to at least 2 edges of color $c(i)$. Then, arguing similarly to Lemma 3, there must be two colors that, when applied to $e$, only lead to one conflict that involves $e$. More formally, let $x$ be the number of colors that appear at least twice. Then if there is no free color the number of edges incident to $e$ must respect $x \cdot 2 + (k - x) \cdot 1 \leq 2\Delta - 2$, which is equivalent to $x \leq \Delta - 2$. With probability at least $1/2$, conflict $i$ will be declared resolved (cf. Section 2.2). The probability for these events is at least $2/(km) \cdot 1/2 = 1/(km)$.

If $e$ is only incident to one edge of color $c(i)$ (the other edge of conflict $i$) then trying to recolor $e$ with color $c(j)$ will be rejected as it would increase the number of conflicts. However, it would swap tags $i$ and $j$ and, consequently, $\varphi_{t+1}(i) = \varphi_t(i) - 1$. The probability for this recolor operation is $1/(km)$. □

We conclude that $\varphi_t$ is dominated by a lazy[2] fair random walk on $\{0, \dots, h\}$ where the probability of changing the current state is at least $1/(km)$. By the first statement of Lemma 27 in the appendix, the expected time to reach state 0 is at most $h^2 km$. Since there are up to $m^2$ fair random walks for all tags (which are not necessarily independent), the third statement of Lemma 27 yields that the expected time for all tags to disappear is $O(h^2 km \log(m^2)) = O(h^2 km \log m)$. □

## 6 TOWARDS AN ANALYSIS OF TOROIDS

We now turn our attention to the performance of RLS on toroids, which are essentially two-dimensional grids with edges "wrapping around". The reason for studying toroids is that they represent a simple graph class featuring many cycles. We will see in the following that cycles play a key role in edge coloring, and that the analysis can become quite involved. We believe that many of the arguments applied to bipartite toroids also apply to general $\Delta$-regular bipartite graphs, or even arbitrary bipartite graphs.

Toroids are formally defined as graphs with vertices $(i, j)$ for $1 \leq i \leq n_1$ and $1 \leq j \leq n_2$ and edges from $(i, j)$ to vertices $(i + 1, j)$, $(i - 1, j)$, $(i, j + 1)$ and $(i, j - 1)$, where for ease of notation we identify indices 0 with $n_1$ and $n_1 + 1$ with 1 for the first argument and likewise for $n_2$ and the second argument. The number of vertices in a toroid is $n_1 \cdot n_2$.

We imagine a toroid drawn as a 2-dimensional grid, with edges wrapping around, such that edges are drawn either horizontally or vertically (see Figure 2 for an example). We speak of rows and columns in an obvious way.

Note that a toroid with parameters $n_1, n_2$ is bipartite if both $n_1$ and $n_2$ are even. We always assume that $n_1, n_2 \geq 4$ as then the toroid is 4-regular, that is, every vertex has degree 4. This implies

[2] The term *lazy* means that the random walk has a positive self-loop probability.

that the number of edges is $m = 2n$. In the following, we tacitly assume that all toroids are 4-regular.

Bipartite toroids are 4-edge-colorable, and the number of proper colorings is exponential. For example, all colorings where rows are colored with alternating colors 1 and 2 (say) and columns are colored with alternating colors 3 and 4, are proper colorings. For each row and column we can choose independently which of the two colors comes first, which gives rise to $2^{n_1+n_2}$ different proper colorings. There are many further proper colorings that do not follow patterns of rows and columns (see Figure 2 for a coloring that is nearly proper). Note that, since $k = 4$ colors are used and toroids are $k$-regular, every vertex in every proper coloring must be incident to exactly one edge of each color. The orientation of these edges can vary between neighboring vertices.

For improper colorings we show that there exist unique paths of alternating colors that start and end in a common vertex of a conflict. We refer to a simple path as *$i$-$j$-path* if colors $i$ and $j$ are alternating on the path.

Lemma 21. *Consider a conflict $(e_1, e_2)$ with color $i$ and common vertex $v$, where $e_1 = \{v_1, v\}$ and $e_2 = \{v, v_2\}$. Then the following statements hold:*

1. *For all colors $j \neq i$, there is a unique $i$-$j$-path that starts at $v$, uses $e_1$ but not $e_2$ and ends in a vertex $w$ that is the common vertex of a conflict. The same holds when the roles of $v_1$ and $v_2$ are swapped.*
2. *For all colors $j \neq i$, the unique $i$-$j$-path starting with $e_1$ does not share any edges with the unique $i$-$j$-path starting with $e_2$.*
3. *All $i$-$j$-paths where $j$ is a free color at $v$ end in a different conflict.*

Proof. We follow this $i$-$j$-path, starting from $v$ and moving to $v_1$. For every vertex $u$ on this path, the following holds. If $u$ has more than one incident edge colored $j$ or $u$ has more than one incident edge colored $i$, $w = u$ and the claim holds. If $u$ only has one incident edge colored $i$ or $j$, by the pigeon-hole principle $u$ must have two incident edges of a different color and we can take $w = u$. If the above cases do not occur, $u$ has exactly one $i$-colored edge and one $j$-colored edge, and the path can be extended, while remaining unique.

The path cannot have any loops, hence it must reach a conflict without using edge $e_2$ or return to $v$ via $e_2$. We show that the latter case is impossible. Assume for a contradiction that it returns to $v$ via $e_2$, closing a cycle. Then the first and the last edge of the path were colored $i$. Since colors must alternate on the path, the cycle must have odd length, contradicting the assumption that the toroid is bipartite. Hence the path must end in a conflict without using $e_2$.

This argument also shows that the unique $i$-$j$-path starting with $e_1$ has no common edges with the unique $i$-$j$-path starting with $e_2$, proving the second statement.

For the third statement, if $j$ is a free color at $v$, there can be no $i$-$j$ paths looping back to $v$ as the last edge cannot be colored $j$ (as $j$ is a free color at $v$) nor $i$ (as it would close an odd cycle). □

Lemma 21 in particular implies that every improper coloring must have at least two conflicts.

The following lemma shows that conflicts can move along $i$-$j$ paths, where $i$ is the color of the conflict and $j$ is a free color at its common vertex. After one such step, the roles of $i$ and $j$ are swapped. A requirement for the lemma to hold is that no other conflicts interfere.

Lemma 22. *Consider a conflict $(e_1, e_2)$ with color $i$ and common vertex $v$, where $e_1 = \{v_1, v\}$ and $e_2 = \{v, v_2\}$. Assume there is no other conflict that has $v$, $v_1$ or $v_2$ as common vertex. Then*

1. *there is a unique free color $j$ at $v$*
2. *the only accepted moves involving the conflict $(e_1, e_2)$ are those where $e_1$ or $e_2$ respectively is recolored $j$*
3. *after such a move is applied, the conflict has color $j$ and $i$ is a free color at its joint vertex.*

Proof. There must be a unique free color at $v$ since $\deg(v) = 4$ and the two remaining edges must have different colors to each other and different from $i$ (as otherwise there would be another conflict with $v$ as common vertex). Let the free color be $j$.

Since there is no conflict with $v_1$ as common vertex, all colors must be present exactly once at $v_1$. The same holds for $v_2$.

If $e_1$ is recolored $j$ then $e_1$ and $e_2$ stop being conflicting, and $e_1$ starts being conflicting with the unique $j$-colored edge at $v$. This is a fitness-neutral move that moves the conflict towards a new joint vertex $v_1$.

If $e_1$ is recolored $s \in \{1, \dots, c\} \setminus \{i, j\}$ then the number of conflicts increases as there is one $s$-colored edge at $v_1$ and another $s$-colored edge at $v$. Hence the only accepted move for $e_1$ is to recolor is with $j$, the free color at $v$.

All the above holds analogously for $e_2$, completing the proof of the second statement.

The third statement holds since, before applying the move, $e_1$ and $e_2$ are the only $i$-colored edges at $v_1$ and $v_2$, respectively. When one of these edges is recolored, $i$ becomes a free color at the respective vertex. □

We also characterize edges that cannot be recolored as they lead to rejected moves.

Lemma 23. *Consider an edge $e$ that is not part of any conflict. If $e$ has an end point where all other colors are present then all local moves recoloring $e$ will be rejected.*

Proof. Let $e = \{u, v\}$ and w. l. o. g. let all other colors be present at $v$. Since $e$ is not part of any conflict, no conflicts will be resolved by recoloring $e$. However, a new conflict will be created with $v$ as common vertex. Thus the number of conflicts will increase and the move will be rejected. □

In the following, we consider the time to resolve the last two remaining conflicts. We show a lower bound of $\Omega(m^3)$ when starting with a particular coloring with just two conflicts. Then we argue why we believe that this bound is asymptotically tight and why this is difficult to prove formally.

Theorem 24. *For every bipartite toroid, there is a search point with just two conflicts from which RLS with $k = 4$ colors needs expected time $\Omega(m^3)$ to find a proper 4-coloring.*

Proof. A cycle is called *chordless* if no two vertices are connected by an edge that does not itself belong to the cycle (the cycle highlighted in Figure 2 is chordless).

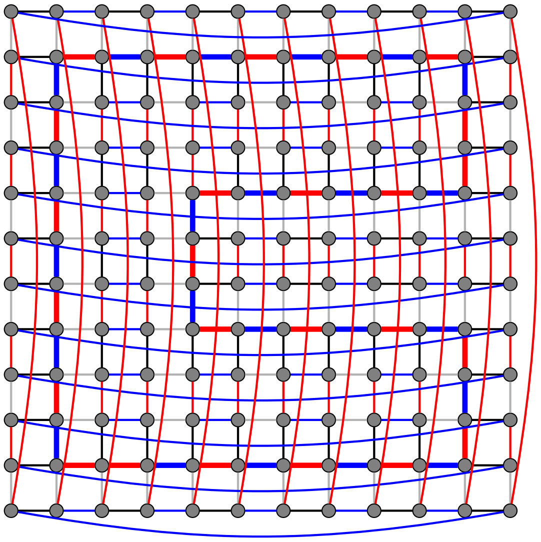

**Figure 2: Sketch of a worst-case initial coloring for toroids. The cycle drawn in bold uses only colors red and blue, with colors alternating, bar two conflicts.**

We construct a coloring with two conflicts lying on a chordless cycle $C$ of length $\Theta(m)$. The colors on the cycle are alternating, bar the two conflicts. The conflicts are placed at an initial distance of $\Theta(m)$. A chordless cycle of length $\Theta(m)$ can be constructed by "snaking" left and right and leaving a safety gap to parts of the cycle that are already constructed. (Taking care when choosing those gaps that the remainder of the graph can still be properly colored.) Figure 2 shows an example. Note that the construction can easily be scaled up for larger graphs by duplicating rows and/or columns appropriately.

Call the common vertices of the two conflicts $v_1$ and $v_2$, respectively. We have two $i$-$j$-paths between $v_1$ and $v_2$ that together form the cycle $C$, where $i$ and $j$ are the colors of the conflicts and the free colors at $v_1$ and $v_2$. We call these paths *augmenting paths* (inspired by well-known algorithms for maximum matchings and subsequent studies of EAs [13]) as swapping colors on all edges of the path yields a fitness improvement (and in this case, a proper coloring).

We shall pay particular attention to the length of the shortest augmenting path. Once this length has reduced to 1, RLS is able to recolor this edge and, if the right color is chosen, this yields a proper coloring. The idea of considering the shortest augmenting path is borrowed from Fischer and Wegener's analysis of coloring problems on cycle graphs [11]. On the cycle $C$, the length of the shortest augmenting path corresponds to the graph distance of $v_1$ and $v_2$ on the subgraph induced by $C$.

By Lemma 23, as long as $v_1$ and $v_2$ are not adjacent on the cycle, only local moves at the conflicting edges will be accepted. This is because the cycle is chordless and $v_1$ and $v_2$ can only be adjacent if they are adjacent on the cycle. All other vertices have edges of all four colors, thus every non-conflicting edge meets the conditions of Lemma 23. In other words, the only accepted moves are those moving one of the conflicts along the cycle, unless the conflicts' common vertices have reached a distance of 1. Once this happens, we pessimistically assume that a proper coloring has been found.

Both conflicts can travel in either direction with equal probability $1/(km)$. This implies that, if the length of the shortest augmenting path is less than $|C|/2$, there are two local moves that reduce this length by 1, and there are two local moves that increase this length by 1. If the maximum possible length of $|C|/2$ is reached[3], there are 4 local moves that decrease the length of the shortest augmenting path.

Hence the process can be regarded as a fair random walk on states $\{1, 2, 3, \ldots, |C|/2\}$ with a reflecting state $|C|/2$ and transition probabilities to neighboring states of $2/(km)$ (and $4/(km)$ in the case of $|C|/2$). With the remaining probability, the random walk stays put.

Since $|C| = \Theta(m)$, the initial distance is $\Theta(m)$, and transitions happen with probability $4/(km)$, by Lemma 27 the expected time to reach state 1 is $\Theta(m^3)$. □

It seems plausible that the last non-optimal fitness level is the most difficult one, in the worst case. We conjecture that the lower bound from Theorem 24 is tight and that the last non-optimal fitness level is optimized in expected time $O(m^3)$ for all colorings with two conflicts remaining. We do not have a formal proof for this conjecture for reasons explained in the following.

Call the common vertices of the two conflicts $v_1$ and $v_2$. By Lemma 21 there are two unique $i$-$j$-paths connecting $v_1$ and $v_2$ that form a cycle $C$. As in the proof of Theorem 24 we consider the length of the shortest augmenting path, or equivalently the graph distance between the conflicts' common vertices on $C$. The proof of Theorem 24 has already established an upper bound of $O(m^3)$ for reaching a state 1, assuming that $C$ is chordless. Note that from state 1 there is a probability of at least $1/(km) = \Omega(1/m)$ of finding the optimum in the next step. There is also a probability of at most $3/m = O(1/m)$ of making any other accepted move (for instance, increasing the current state) as in this situation, by Lemma 23, only moves affecting one of the 3 edges that are part of a conflict may be accepted. Hence, there is a constant probability that the optimum will be found within the next $O(m)$ steps before any other accepted move is made. If this is not the case, we repeat the above arguments. Thus, if suffices to bound the expected time to reach state 1 by $O(m^3)$.

A problem arises if $C$ is not chordless and if $v_1$ and $v_2$ are connected by an edge not on $C$. Let $a \notin \{i, j\}$ denote the color of $\{v_1, v_2\}$. Both conflicts must have the same color $i$ as otherwise every path between $v_1$ and $v_2$ would have even length and the edge $\{v_1, v_2\}$ would close an odd cycle. But in this situation the edge $\{v_1, v_2\}$ can be recolored $j$ in a fitness-neutral operation as $j$ is a free color for both $v_1$ and $v_2$ (note that Lemma 23 does not apply). This means that the free color at both $v_1$ and $v_2$ switches from $j$ to $a$, and there is a corresponding cycle $C'$ with alternating colors $i$-$a$ between $v_1$ and $v_2$ on which the conflicts are able to move. Note that the colors may switch back to $i$ and $j$ at $\{v_1, v_2\}$[4], but the colors might also switch again towards arbitrary combinations of colors on further cycles $C''$, $C'''$, and so on.

[3] Note that $|C|/2$ is an integer as $C$ must be of even length.

[4] In other words, if we consider the state graph of all possible colorings that can be reached via fitness-neutral local moves, that graph is undirected.

The same effect may happen even in chordless cycles when the distance of the shortest augmenting path has reduced to 1. Then the edge $\{v_1, v_2\}$ is incident to two edges of the two colors different from $i$ and $j$. Recoloring the edge with such a color is a fitness-neutral move as it removes the two conflicts on $C$, while creating two new conflicts with joint vertices $v$ and $w$. This switches the random walk to another cycle $C'$, while the length of the shortest augmenting path remains at 1. Even if $C$ was chordless, $C'$ may not be chordless. Hence, to get a rigorous upper bound of $O(m^3)$ for the last fitness level, we would have to assume that all cycles that can ever be reached are chordless.

The situation is complicated further when more than two conflicts are present. Other conflicts may interfere with the process described above in various ways:

- They can block augmenting paths at one end. While this is the case (and no other interference happens), one end of the augmenting path will be fixed, while the other end can perform a random walk. Then the previous random walk arguments can still be applied with transition probabilities reducing from $2/(cm)$ to $1/(cm)$.
- Augmenting paths may become blocked at both ends, in which case they cease to be "augmenting". For trees we used the idea of tags being swapped, so that tags could roam more freely even though the original conflicts were being blocked. Tags could always be removed when reaching leaves. It is not clear whether or how this idea can be applied for toroids as we are lacking conditions on when tags will disappear.
- If two conflicts share the same common vertex $v$, there are two free colors at $v$, possibly increasing the number of augmenting paths.
- Augmenting paths that are blocked can become unblocked, which may suddenly and drastically increase the length of the shortest augmenting path.

It seems plausible that search points with many conflicts have many augmenting paths. However, proving this does not seem obvious, even when we consider paths that are blocked on exactly one end as augmenting paths. Even proving that a single augmenting path exists is not obvious. It is possible to construct colorings where several conflicts all block each other on both ends. Hence, it is an open problem to prove or disprove that in every improper coloring there exist conflicts that are not blocked on both ends.

Note, however, that even if all conflicts end up being blocked, it may still be likely that conflicts become unblocked once other conflicts have moved about. And all the above considerations arise from a worst-case perspective, and trying to prove statements that apply to *every* improper coloring. Observing simulations suggests that blocked conflicts does not seem to be a real issue for performance. In all runs observed, RLS found a proper coloring in a time that seems close to a function $am^3$ for a small constant $a$ (see Section 7). We therefore formulate the following conjecture for future work.

Conjecture 25. *For every bipartite toroid $G$ and every initial coloring, RLS finds a proper 4-coloring in expected time $O(m^3)$.*

## 7 EXPERIMENTS

In the following we supplement our theoretical findings with extensive experimentation. We consider all graph classes analyzed in the foregoing sections: paths, even cycles, star graphs, binary trees as a special case of trees and toroidal graphs with dimensions $n_1 = n_2 = \sqrt{n}$ and $\sqrt{n}$ an even integer. Additionally, we consider complete graphs $K_n$ with even $n$, complete bipartite graphs $K_{n/2,n/2}$ with equally sized partitions and $d$-dimensional hypercubes as special cases of $\Delta$-regular graphs. It seems natural to consider the number of edges $m$ as an upper limit for the size of the graphs. Here, we perform experiments for graphs with at most $m = 512$ edges. Note that this allows values of $m \in \{4, 12, 32, 80, 192, 448\}$ for hypercubes, but $m \in \{1, 2, \dots, 512\}$ for, e. g., paths. For reasons of comparability and to keep the computational effort justifiable we take the values for the hypercube as the baseline and consider similar values for all other graph classes. For statistical soundness we perform 50 independent runs on each graph instance for both RLS and (1+1) EA and measure the number of function evaluations until a proper coloring with $\chi'(G)$ colors is generated for the first time. Plots of the average running times of RLS[5] and fitted regression models with 95% confidence intervals are depicted in Figure 3. Accompanying results of the regression analysis are provided in Table 2.[6] A visual inspection of the fitted regression curves reveals that the models seem to fit the data very well. This observation is supported by the $R^2$ indicator and the root mean squared error (RMSE). While the former measures the fraction of variation in the data explained by the model (the closer to 1 the better), the latter describes the average deviation of predicted values and actual observations from the data. The $R^2$ values are $\geq 0.99$ for all trained models, indicating a very good fit. This is supported by the low RMSE values (relative to the potential range of fitness evaluations for the corresponding graph class).

Moreover, we observe a clear pattern in the quotient of the estimated model coefficients $a$ for (1+1) EA and RLS which are all very close to $e \approx 2.71$. Since $e$ reflects the waiting time for (1+1) EA to perform a single local move, this suggests that (1+1) EA is most effective when only recoloring a single edge.

In summary, the experimental study supports all theoretical results obtained in this paper.

In all graphs studied here, the runtime was bounded by, or is conjectured to be bounded by $O(\Delta\ell^2 m \log m)$. (In some cases, such as cycles, paths, star graphs or potentially toroids, the $\log m$ factor may be dropped.) The experiments gave further strong evidence for this bound for further graph classes, including hypercubes, complete graphs and complete bipartite graphs. In all cases we obtained a very good fit with functions $a\Delta\ell^2 m \log m$ or $a\Delta\ell^2 m$ with very reasonable leading constants $a$. Again, the model suitability is supported by $R^2$-values close to 1 and very low RMSE. In fact, in particular for complete bipartite graphs and all interesting special cases of bipartite graphs, i. e., toroids, complete binary trees and hypercubes, RMSE values are negligible and the model fit is almost perfect. We hence state the following conjecture for future work.

Conjecture 26. *RLS and (1+1) EA find a proper $\Delta$-coloring for every bipartite graph $G$ with maximum degree $\Delta$ and $\ell := \ell(G)$ in expected time $O(\Delta\ell^2 m \log m)$.*

[5] We do not show plots for (1+1) EA since they do not reveal any more information.

[6] For regression analysis the statistical programming language R [23] (version 3.5.2) was used. In particular we used the `lm(...)` function to fit regression models.

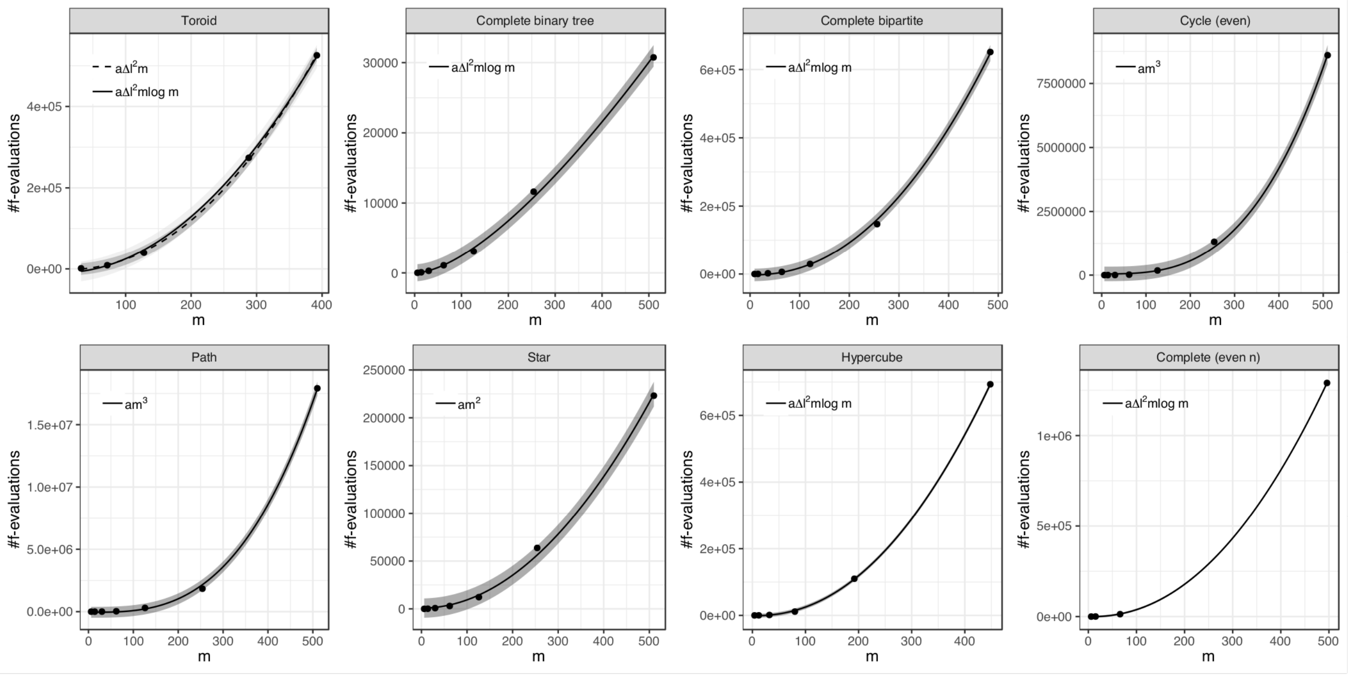

**Figure 3: Average runtime of RLS (black dots) and fitted regression functions with 95% confidence intervals separated by graph classes.**

**Table 2: Results of a regression analysis with different regression models for RLS and (1+1) EA for optimal edge-colorings on different graph classes.**

| | | **RLS** | | | **(1+1) EA** | | | |
|---|---|---|---|---|---|---|---|---|
| **Graph class** | **Model** | **Coeff. *a*** | **$R^{2*}$** | **RMSE$^\dagger$** | **Coeff. *a*** | **$R^2$** | **RMSE** | **Coeff. ratio$^\ddagger$** |
| Toroid | $a\Delta\ell^2 m$ | 3.4644 | 0.9992 | 5064.7 | 10.6784 | 0.9973 | 28286.2 | 3.082 |
| | $a\Delta\ell^2 m \log m$ | 0.5779 | 0.9997 | 2979.8 | 1.7822 | 0.9994 | 13721.1 | 3.084 |
| Complete bin. tree | $a\Delta\ell^2 m \log m$ | 0.2504 | 0.9986 | 365.2 | 0.6278 | 0.9988 | 829.5 | 2.507 |
| $K_{n/2,n/2}$ | $a\Delta\ell^2 m \log m$ | 0.4494 | 0.9993 | 5545.1 | 1.2033 | 0.9999 | 4924.0 | 2.678 |
| Cycle (even) | $am^3$ | 0.0647 | 0.9990 | 86286.4 | 0.1555 | 0.9999 | 60392.9 | 2.403 |
| Path | $am^3$ | 0.1352 | 0.9994 | 132298.6 | 0.3403 | 0.9998 | 194497.0 | 2.517 |
| Star | $am^2$ | 0.8612 | 0.9981 | 3047.9 | 2.3638 | 0.9982 | 8234.7 | 2.745 |
| Hypercube | $a\Delta\ell^2 m \log m$ | 0.5667 | 1.0000 | 1591.2 | 1.5923 | 1.0000 | 2456.4 | 2.810 |
| $K_n$ ($n$ even) | $a\Delta\ell^2 m \log m$ | 0.8455 | 1.0000 | 804.3 | 2.1024 | 1.0000 | 1138.7 | 2.487 |

$^*$ $R^2$: Fraction of variance explained by model; $^\dagger$ RMSE: Root Mean Squared Error; $^\ddagger$ Quotient of regression coefficients of (1+1) EA and RLS

## 8 CONCLUSIONS

We have presented the first runtime analysis of evolutionary algorithms on the edge coloring problem, for which it is $\mathcal{NP}$-hard to decide whether $\Delta$ or $\Delta + 1$ edge colors are sufficient. We presented general results on the time to obtain $(2\Delta - 1)$-colorings, reducing the number of conflicts down to $m$ and two lower bounds that apply to all connected graphs. For cycles, paths, star graphs and arbitrary trees we have shown that simple evolutionary algorithms such as

RLS and (1+1) EA are able to find proper colorings with a minimum number of $\Delta$ colors efficiently, for all initial colorings (see Table 1 for details).

We then considered toroids as a graph class with many cycles, where the analysis of RLS turned out to be surprisingly complex. We presented a lower bound on the expected time to resolve the final two conflicts, starting from a worst-case initial coloring with two conflicts. Then we discussed the challenges involved in proving rigorous upper bounds for the time to resolve the last two conflicts, and for analysing dynamics with more than two conflicts. Experiments support our conjecture that RLS finds proper $\Delta$-colorings on bipartite toroids in expected time $O(m^3)$.

More generally, both theory and experiments support the conjecture that RLS can find proper $\Delta$-colorings on all bipartite graphs in expected time $O(\Delta\ell^2 m \log m)$.

Avenues for future work include proving the above conjectures and finding graphs that are hard to color optimally for RLS. We know that such graphs must exist as edge coloring is $\mathcal{NP}$-hard. However, RLS and (1+1) EA performed well on all graph classes that were so far considered theoretically and/or experimentally.

## ACKNOWLEDGMENTS

The authors thank Thomas Sauerwald for insightful discussions and references on random walks. Jakob Bossek acknowledges funding from COST Action CA15140[7]: *Improving Applicability of Nature-Inspired Optimisation by Joining Theory and Practice* (ImAppNIO).

## REFERENCES

[1] Noga Alon. 2003. A Simple Algorithm for Edge-coloring Bipartite Multigraphs. *Inf. Process. Lett.* 85, 6 (2003), 301–302.

[2] Anne Auger and Benjamin Doerr (Eds.). 2011. *Theory of Randomized Search Heuristics – Foundations and Recent Developments.* Number 1 in Series on Theoretical Computer Science. World Scientific.

[3] Leonid Barenboim and Michael Elkin. 2013. *Distributed Graph Coloring: Fundamentals and Recent Developments.* Morgan & Claypool.

[4] Leonid Barenboim, Michael Elkin, and Tzalik Maimon. 2017. Deterministic Distributed $(\Delta + o(\Delta))$-Edge-Coloring, and Vertex-Coloring of Graphs with Bounded Diversity. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC '17).* ACM, 175–184. https://doi.org/10.1145/3087801.3087812

[5] Jakob Bossek, Frank Neumann, Pan Peng, and Dirk Sudholt. 2019. Runtime Analysis of Randomized Search Heuristics for Dynamic Graph Coloring. In *Proceedings of the 21th Annual Genetic and Evolutionary Computation Conference (GECCO '19).* ACM, Prague, Czech Republic. https://doi.org/10.1145/3321707.3321792

[6] Yi-Jun Chang, Qizheng He, Wenzheng Li, Seth Pettie, and Jara Uitto. 2018. The Complexity of Distributed Edge Coloring with Small Palettes. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '18).* SIAM, 2633–2652.

[7] Raymond Chiong, Thomas Weise, and Zbigniew Michalewicz (Eds.). 2012. *Variants of Evolutionary Algorithms for Real-World Applications.* Springer.

[8] Richard Cole, Kirstin Ost, and Stefan Schirra. 2001. Edge-Coloring Bipartite Multigraphs in $O(E \log D)$ Time. *Combinatorica* 21, 1 (2001), 5–12.

[9] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. 2012. Multiplicative Drift Analysis. *Algorithmica* 64, 4 (01 Dec 2012), 673–697.

[10] Thomas Erlebach and Klaus Jansen. 2001. The Complexity of Path Coloring and Call Scheduling. *Theoretical Computer Science* 255, 1 (2001), 33–50. https://doi.org/10.1016/S0304-3975(99)00152-8

[11] Simon Fischer and Ingo Wegener. 2005. The One-dimensional Ising Model: Mutation versus Recombination. *Theoretical Computer Science* 344, 2–3 (2005), 208–225.

[12] Mohsen Ghaffari, Fabian Kuhn, Yannic Maus, and Jara Uitto. 2018. Deterministic Distributed Edge-coloring with Fewer Colors. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018).* ACM, 418–430. https://doi.org/10.1145/3188745.3188906

[13] Oliver Giel and Ingo Wegener. 2003. Evolutionary Algorithms and the Maximum Matching Problem. In *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS '03).* Springer-Verlag, Berlin, Heidelberg, 415–426.

[14] Ian Holyer. 1981. The NP-Completeness of Edge-Coloring. *SIAM J. Comput.* 10, 4 (1981), 718–720. https://doi.org/10.1137/0210055

[15] Thomas Jansen. 2013. *Analyzing Evolutionary Algorithms – The Computer Science Perspective.* Springer.

[16] Daniel Johannsen. 2010. *Random Combinatorial Structures and Randomized Search Heuristics.* Ph.D. Dissertation. Universität des Saarlandes, Saarbrücken, Germany and the Max-Planck-Institut für Informatik.

[17] Johannes Lengler. 2017. Drift Analysis. *CoRR* (2017). http://arxiv.org/abs/1712.00964

[18] Nancy A. Lynch. 1996. *Distributed Algorithms.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[19] Jayadev Misra and David Gries. 1992. A Constructive Proof of Vizing's Theorem. *Inform. Process. Lett.* 41, 3 (1992), 131–133. https://doi.org/10.1016/0020-0190(92)90041-S

[20] Frank Neumann. 2008. Expected Runtimes of Evolutionary Algorithms for the Eulerian Cycle Problem. *Computers & Operations Research* 35, 9 (2008), 2750–2759.

[21] Frank Neumann and Ingo Wegener. 2004. Randomized Local Search, Evolutionary Algorithms, and the Minimum Spanning Tree Problem. In *Proceedings of the 6th Annual Genetic and Evolutionary Computation Conference (GECCO '04).* Springer Berlin Heidelberg, 713–724.

[22] Frank Neumann and Carsten Witt. 2010. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity.* Springer.

[23] R Core Team. 2018. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

[24] Jens Scharnow, Karsten Tinnefeld, and Ingo Wegener. 2005. The Analysis of Evolutionary Algorithms on Sorting and Shortest Paths Problems. *Journal of Mathematical Modelling and Algorithms* 3, 4 (2005), 349–366. https://doi.org/10.1007/s10852-005-2584-0

[25] Dirk Sudholt. 2005. Crossover is Provably Essential for the Ising Model on Trees. In *Proceedings of the 7th Annual Genetic and Evolutionary Computation Conference (GECCO '05).* ACM, 1161–1167.

[26] Dirk Sudholt and Christine Zarges. 2010. Analysis of an Iterated Local Search Algorithm for Vertex Coloring. In *Algorithms and Computation.* Springer Berlin Heidelberg, 340–352.

[27] David P Williamson, Leslie A Hall, J A Hoogeveen, Cor A J Hurkens, Jan Karel Lenstra, Sergey Vasil'evich Sevast'janov, and David B Shmoys. 1997. Short Shop Schedules. *Operations Research* 45, 2 (1997), 288–294. https://doi.org/10.1287/opre.45.2.288

[28] Carsten Witt. 2005. Worst-case and Average-case Approximations by Simple Randomized Search Heuristics. In *Proceedings of the 22nd Annual Conference on Theoretical Aspects of Computer Science (STACS'05).* Springer-Verlag, Berlin, Heidelberg, 44–56.

# A RANDOM WALK TOOLS

The following results on fair random walks are folklore and/or follow from standard arguments. We gather them here as we are not aware of a reference presenting these statement in this form. The lemma is used in the main part and may be of future use.

Lemma 27. *Consider a fair random walk $X_t$ on $\{0, \dots, k\}$ where 0 is an absorbing state and $k$ is a reflecting state. More formally, abbreviating $p_{i,j} := \Pr(X_{t+1} = j \mid X_t = i)$, for all $0 < i < k$, $p_{i,i+1} = p_{i,i-1} = 1/2$, $p_{0,0} = 1$ and $p_{k,k-1} = 1$. Let $T$ be the first hitting time of state 0. Then the following statements hold:*

1. *For all $X_0$, $E(T \mid X_0) = X_0(2k - X_0 - 1) < k^2$.*
2. *For all $X_0$ and all $r \in \mathbb{N}$, $\Pr\left(T \geq 2rk^2 \mid X_0\right) \leq 2^{-r}$.*
3. *Consider $s > 1$ not necessarily independent random walks with the given transition probabilities. Let $T^{(s)}$ denote the time for all $s$ random walks to hit state 0. Then $E\left(T^{(s)}\right) = O(k^2 \log s)$.*

*All statements also hold for a lazy random walk with a self-loop probability of $1 - p$, when multiplying all time bounds by $1/p$.*

Proof. The first statement follows from the following folklore argument. Imagine a fair random walk $X'_t$ on a state space $\{0, \dots, k, \dots, 2k - 1\}$ where states 0 and $2k - 1$ are both absorbing.

[7] https://www.cost.eu/actions/CA15140/

Now, for every $0 \leq i \leq k-1$, state $i$ is identified with state $2k-1-i$. Then $X_t'$ is identical to $X_t$, but the reflecting state $k$ has been replaced by an absorbing state $2k-1$. Now gambler's ruin, applied to $X_t'$ with an initial state of $X_0$, yields $\mathrm{E}(T \mid X_0) = X_0(2k - X_0 - 1)$. The right-hand side is at most $k^2$.

The second statement follows from standard arguments on independent phases. By the first statement and Markov's inequality, $\Pr\left(T \geq 2k^2\right) \leq 1/2$, irrespective of the initial state $X_0$. Consider $r$ phases, each of $2k^2$ subsequent steps, then the probability that state 0 will be missed in all $r$ phases is $2^{-r}$.

The third statement follows from applying the second statement with $r := \log(s) + 1$. This implies that a fixed random walk will not have hit state 0 with probability at most $2^{-r} = 1/(2s)$ after a period of $2rk^2$ steps. Taking a union bound over all $s$ random walks, the probability that one of them will not have hit state 0 is at most $1/2$. In this case we reiterate the above arguments with another period of $2rk^2$ steps. In expectation, only 2 periods are needed, hence $\mathrm{E}\left(T^{(s)}\right) \leq 4rk^2 = O(k^2 \log s)$.

For the lazy random walk, the first statement still holds as the expected waiting time for a transition is $1/p$, thus $\mathrm{E}(T \mid X_0) = X_0(2k - X_0 - 1)/p$. The applications of Markov's inequality and the union bound in the proofs of the second and third statements, respectively, remain unaffected when introducing a factor of $1/p$ appropriately. □