# Concentration Tail-Bound Analysis of Coevolutionary and Bandit Learning Algorithms*

**Per Kristian Lehre , Shishen Lin**
University of Birmingham, Birmingham, United Kingdom
{p.k.lehre, sxl1242}@cs.bham.ac.uk

## Abstract

Runtime analysis, as a branch of the theory of AI, studies how the number of iterations algorithms take before finding a solution (its runtime) depends on the design of the algorithm and the problem structure. Drift analysis is a state-of-the-art tool for estimating the runtime of randomised algorithms, such as evolutionary and bandit algorithms. Drift refers roughly to the expected progress towards the optimum per iteration. This paper considers the problem of deriving concentration tail-bounds on the runtime/regret of algorithms. It provides a novel drift theorem that gives precise exponential tail-bounds given positive, weak, zero and even negative drift. Previously, such exponential tail bounds were missing in the case of weak, zero, or negative drift.

Our drift theorem can be used to prove a strong concentration of the runtime/regret of algorithms in AI. For example, we prove that the regret of the RWAB bandit algorithm is highly concentrated, while previous analyses only considered the expected regret. This means that the algorithm obtains the optimum within a given time frame with high probability, i.e. a form of algorithm reliability. Moreover, our theorem implies that the time needed by the coevolutionary algorithm RLS-PD to obtain a Nash equilibrium in a BILINEAR max-min-benchmark problem is highly concentrated. However, we also prove that the algorithm forgets the Nash equilibrium, and the time until this occurs is highly concentrated. This highlights a weakness in the RLS-PD which should be addressed by future work.

## 1 Introduction

Drift analysis is a powerful technique in understanding the performance of randomised algorithms, particularly in the field of runtime analysis of heuristic search. For more recent overviews of drift analysis in evolutionary computation, see [Doerr and Neumann, 2019; Neumann and Witt, 2010; Jansen, 2013; He and Yao, 2001; Hajek, 1982]. The majority of existing drift theorems provide an upper bound on the expected runtime needed to reach a target state, such as an optimal solution set [He and Yao, 2001]. By identifying an appropriate potential function and demonstrating a positive drift towards the target state, the expected runtime can be bounded by the reciprocal of the drift multiplied by the maximum distance from the target state.

The focus of concentration tail-bound analysis is on quantifying the deviation of the runtime of randomised algorithm $T$, from its expected value. By providing insights into the distribution of $T$, this approach offers a more detailed understanding of an algorithm's performance [Kötzing, 2016; Lehre and Witt, 2021; Doerr and Goldberg, 2013]. The concentration tail-bound analysis has gained significant interest due to its potential for delivering tighter upper bounds on the runtime of various algorithms. For instance, an exponential tail bound is used to bound the expected runtime of RLS on separable functions [Doerr *et al.*, 2013]. Moreover, in the case of (1+1)-cooperative co-evolutionary algorithms, concentration tail-bound analysis can establish a $\Theta(n\log(n))$ bound for the runtime of the cooperative coevolutionary algorithm on linear functions [Lehre and Lin, 2023]. The concentration tail-bound analysis is also useful in the context of restarting arguments; for example [Case and Lehre, 2020]. More precise runtime estimation can be valuable in optimising and comparing different algorithms, potentially leading to improved algorithm design and performance [Bian *et al.*, 2020; Dang *et al.*, 2021; Zheng *et al.*, 2022; Doerr and Qu, 2023].

Concentration tail bounds are not only used in runtime analysis to help us understand evolutionary algorithms, including simple genetic algorithms or coevolutionary algorithms [Kötzing, 2016; Lehre and Lin, 2023], but can also be used in regret analysis of reinforcement learning algorithms. A typical example is using concentration inequality (Azuma-Hoeffding inequality) to provide precise bounds for regret. Concentration inequalities are used in the development of optimal UCB family algorithms, incorporating the concept of optimism in the face of uncertainty [Auer *et al.*, 2002].

### 1.1 Related Work

Researchers use drift analysis to analyse not only the runtime of evolutionary algorithms but also other randomised

*Full version at https://arxiv.org/abs/2405.04480.

algorithms like Random 2-SAT [Göbel *et al.*, 2022] or the expected regret of simple reinforcement learning algorithms on bandit problems [Larcher *et al.*, 2023]. We would like to explore more advanced drift analysis tools to provide more precise estimates of runtime and regret. Various extensions of drift theorems have been proved, including multiplicative drift [Doerr *et al.*, 2010], variable drift [Baritompa and Steel, 1996; Johannsen, 2010; Mitavskiy *et al.*, 2009], and negative drift [Oliveto and Witt, 2012]. The multiplicative drift theorem [Doerr *et al.*, 2010] refines the original additive drift theorem by considering the current state, resulting in a more precise bound when using the same potential function. Variable drift [Baritompa and Steel, 1996; Johannsen, 2010; Mitavskiy *et al.*, 2009] generalises the multiplicative drift concept to incorporate an increasing positive function, $h$. On the other hand, negative drift [Oliveto and Witt, 2012] is employed to provide a lower bound for the expected runtime, often used to demonstrate that an algorithm has an exponential expected runtime, thereby proving its inefficiency.

In recent years, researchers have been exploring advanced drift theorems that focus on the tail bound of the runtime [Kötzing, 2016; Lehre and Witt, 2021; Doerr and Goldberg, 2013]. Researchers are also interested in the applications of concentration inequalities, like the Azuma-Hoeffding inequalities [Azuma, 1967]. They provide deeper insights into the behaviour and performance of randomised algorithms [Doerr and Künnemann, 2013].

### 1.2 Our Contributions

This paper provides a novel perspective on analysing tail bounds by introducing a classic recurrence strategy. With the help of the recurrence strategy, this paper presents a sharper bound for all possible drift cases with a simpler proof. In particular, we provide an exponential tail bound under constant variance with negative drift. Refining an existing method, we also show a more precise exponential tail bound for the traditional cases with additive drift and for the cases in which there is constant variance but weak or zero drift.

Finally, we illustrate the practical impact of our findings by applying our theorems to various algorithms. The analysis brings us stronger performance guarantees for these algorithms. In particular, we prove the instability of the co-evolutionary algorithm (CoEA) on maximin optimisation (BILINEAR problem instance) occurs with high probability. Moreover, we show that the randomness of the reinforcement learning algorithm RWAB can help to find the optimal policy for the 2-armed non-stationary bandit problem with high probability. This paper is the first tail-bound analysis of both random local search with pairwise dominance (RLS-PD) and the bandit learning algorithm RWAB.

## 2 Preliminaries

For a filtration $\mathcal{F}_t$, we write $\mathrm{E}_t(\cdot) := \mathrm{E}(\cdot \mid \mathcal{F}_t)$. We denote the 1-norm as $|z|_1 = \sum_{i=1}^n z_i$ for $z \in \{0,1\}^n$ and $\mathbb{1}_E$ by indicator function, i.e. $\mathbb{1}_E = 1$ if event $E$ holds and 0 otherwise. With high probability," will be abbreviated as "w.h.p.". We say an event $E_n$ with problem size $n \in \mathbb{N}$ occurs w.h.p. if $\Pr(E_n) \geq 1 - 1/\mathrm{poly}(n)$. We defer pseudo-codes of algorithms and tables in the appendix.

We define the $k$-th stopping time, also called $k$-th hitting time, which will be used in later proofs.

**Definition 1.** ($k$-th stopping time) Given a stochastic process $(X_t)_{t\geq 0}$ on a state space in $\mathbb{R}$. Let the target set $A$ be a finite non-empty subset of $\mathbb{R}$, and then for any $k \geq 0$, we define $T_k = \min\{t \geq k \mid X_t \in A\}$. In particular, $T_0$ is the first hitting time at $A$.

We first provide a formal definition of variance-dominated and variance-transformed processes.

**Definition 2.** (Variance-dominated processes) A sequence of random variables $X_0, X_1, \cdots \in [0, n]$ is a variance-dominated process with respect to the filtration $\mathcal{F}_0, \mathcal{F}_1, \ldots$ if for all $t \in \mathbb{N}$, the following conditions hold:

(1) $\mathrm{E}(X_{t+1} - X_t \mid \mathcal{F}_t) \geq 0$;

(2) $\exists \delta > 0$ such that $\mathrm{E}\big((X_{t+1} - X_t)^2 \mid \mathcal{F}_t\big) \geq \delta$.

**Definition 3.** (Variance-transformed processes) A sequence of random variables $X_0, X_1, \cdots \in [0, n]$ is a variance-transformed process with respect to the filtration $\mathcal{F}_0, \mathcal{F}_1, \ldots$ if for all $t \in \mathbb{N}$, the following conditions hold:

(1) $0 > \mathrm{E}(X_{t+1} - X_t \mid \mathcal{F}_t) \geq -\frac{c}{n}$;

(2) $\exists \delta > 0$ such that $\mathrm{E}\big((X_{t+1} - X_t)^2 \mid \mathcal{F}_t\big) \geq \delta$.

This paper mainly focuses on random processes which consist of positive, weak (almost zero) or even a small negative drift with a constant second moment since these processes exhibit more complicated dynamics [Kötzing *et al.*, 2015; Friedrich *et al.*, 2016; Göbel *et al.*, 2022; Doerr and Zheng, 2020]. A general polynomial tail bound is provided for these in [Kötzing, 2016], but any general exponential tail bounds for these processes are still missing.

In the following sections, we exploit the Optional Stopping Time Theorem to obtain our exponential tail bound. This theorem is crucial for proving the original additive drift theorem, as highlighted by [He and Yao, 2001]. This recurrence method can provide a different perspective to derive the exponential tail bound in runtime analysis. We defer the statements of Optional Stopping Time Theorems in the appendix.

### 2.1 Previous Works and Discussion

With the development of runtime analysis, researchers have established several concentration tail bounds for EAs. For example, [Lehre and Witt, 2021] provides an exponential tail bound for the basic (1+1)-EAs on OneMax functions, which is a well-studied benchmark function to analyse the performance of EAs. To the best of our knowledge, the current best general tail bounds for both processes under the additive drift and variance-dominated processes can be found in [Kötzing, 2016; Lehre and Witt, 2021].

[Kötzing, 2016] shows that the runtime is at most quadratic in $n$ with probability $1 - p$ for any $p > 0$. If we replace $1/p^{\ell \log(c)}$ by $r > 0$ and rewrite it in terms of an upper tail bound, then the original bound becomes that given two constants $1 \leq c < n, \ell > 0$ and for any $r > 0$,

$$\Pr\left(T \geq rn^2\right) \leq \left(\frac{1}{r}\right)^{1/\ell \log(c)}. \tag{1}$$

Although we have established a tail bound for variance-dominated processes that concentrate on the expectation in a polynomial order with respect to $r$ in Equation (1), it is worth exploring whether a more precise concentration tail bound can be derived for such processes, such as an exponential tail. A sharper exponential tail bound can improve the expected runtime estimation and thus provide useful insights into randomised algorithms.

## 3 A Recurrent Method in Upper Tail Bound

Next, we explore how to derive a general framework for providing exponential tail bound for randomised algorithms, including evolutionary algorithms which satisfy certain conditions. We explore the exit time of $X_t$ out of some interval $[0, b]$, using the same set-up as [Kötzing, 2016].

In the proof of McDiamid inequality, [McDiarmid, 1989] also uses the Hoeffding lemma and conditions on the past events to establish the recurrence. [Doerr and Goldberg, 2013] uses the multiplicative drift condition directly to build up the exponential recurrence relation and hence obtain an exponential tail bound. We want to borrow these ideas to derive an exponential tail bound for variance-dominated processes. To do this, we introduce the $k$-th hitting time of the target state, which is also used in the theorem (Theorem 2.6.2) of [Menshikov *et al.*, 2016]. We combine this recurrent method with the extended Optional Stopping Time theorem.

### 3.1 Variance Overcomes Negative Drift W.H.P.

We proceed to prove our main theorem by considering the most general variance drift theorem which overcomes some negative drift. Following the setting of [Hevia Fajardo *et al.*, 2023], in variance-transformed cases, we focus on the first hitting time of a discrete-time stochastic process $X_t$ at 0 given that $X_t \in [0, b]$. The proof of Theorem 1 uses Lemma 1.

**Lemma 1.** *Let $(X_t)_{t\geq 0}$ be random variables over $\mathbb{R}_{\geq 0}$, each with finite expectation. Let $T$ be any stopping time of $X_t$. If there exist constants $r, \eta > 0$ with respect to $j, t$ such that for any $j \geq 0$, $\mathrm{E}\big(\mathbb{1}_{\{T>t\}}\mathbb{1}_{\{|X_t - X_{t+1}| \geq j\}} \mid \mathcal{F}_t\big) \leq r/(1+\eta)^j$, then there exists a positive constant $c$ such that $\mathrm{E}\big(|X_{t+1} - X_t| \cdot \mathbb{1}_{\{T>t\}} \mid \mathcal{F}_t\big) \leq c$ for all $t \in \mathbb{N} \cup \{0\}$.*

By using Lemma 1, we now satisfy condition (4) in the Optional Stopping Time Theorem, enabling us to proceed with the main proof (details in the appendix). Utilising the extended Optional Stopping Time Theorem, we follow a classic approach for generalisation, which also frees us from the fixed step size condition and the need for the Azuma-Hoeffding inequality for sub-Gaussian supermartingales. [1] We further construct a new stochastic process $Y_t = b^2 - (b - X_t)^2 + \delta t$, connected to the original process and the variance of the drift. By employing the idea of the $k$-th hitting time from [Menshikov *et al.*, 2016], we obtain the upper bound for the $k$-th hitting time, allowing us to construct the recurrence. We present the main theorem of this paper.

[1] The proofs in [Kötzing, 2016] mainly rely on Azuma-Hoeffding inequality for sub-Gaussian supermartingales.

**Theorem 1.** *Let $(X_t)_{t\in\mathbb{N}}$ be a sequence of random variables in a finite state space $S \subseteq \mathbb{R}$ adapted to a filtration $(\mathcal{F}_t)_{t\in\mathbb{N}}$, and let $T = \inf\{t \geq 0 \mid X_t \leq 0\}$. Suppose*

*(A1) there exist $\delta > 0$ such that for all $t < T$, it holds that*

$$\mathrm{E}_t\left((X_{t+1} - X_t)^2 - 2(X_{t+1} - X_t)(b - X_t)\right) \geq \delta$$

*(A2) and for all $t \leq T$, it holds that $0 \leq X_t \leq b$.*

*Moreover, for all $t \geq 0$, assume there exist constants $r, \eta > 0$ with respect to $j, t$, for any $j \geq 0$, $\mathrm{E}\big(\mathbb{1}_{\{T>t\}}\mathbb{1}_{\{|X_t - X_{t+1}| \geq j\}} \mid \mathcal{F}_t\big) \leq r/(1+\eta)^j$. Then, for $\tau > 0$, $\Pr(T > \tau) \leq e^{-\tau\delta/eb^2}$.*

Our main result (Theorem 1) allows the increased tolerance of negative drift rather than non-negative drift tendency and only necessitates a constant second moment of drift, instead of variance, as outlined in [Kötzing, 2016]. Consequently, we can establish a more precise exponential tail bound for stochastic processes with a constant second moment of the drift, even under weak, zero, or negative drift.

### 3.2 Standard Variance Drift

This section presents the standard variance drift scenario (Theorem 2) as a corollary of Theorem 1. More precisely, we now restrict to the non-negative drift tendency. We first define several conditions which will be used later.

(C1*) There exist constants $r, \eta > 0$ with respect to $j, t$, such that for any $j \geq 0$ and for all $t \geq 0$,

$$\mathrm{E}\big(\mathbb{1}_{\{T>t\}}\mathbb{1}_{[|X_t - X_{t+1}| \geq j\}} \mid \mathcal{F}_t\big) \leq \frac{r}{(1+\eta)^j}.$$

(C1) There exists a constant $c > 0$ such that $|X_t - X_{t+1}| < c$ for all $t \geq 0$.

(C2) $\mathrm{E}(X_{t+1} - X_t \mid \mathcal{F}_t) \geq 0$ for all $t \geq 0$.

(C3) There exists some constant $\delta > 0$ such that $\mathrm{E}\big(X_{t+1} - X_t)^2 \mid \mathcal{F}_t\big) \geq \delta$ for all $t \geq 0$.

**Theorem 2.** *Let $(X_t)_{t\geq 0}$ be random variables over $\mathbb{R}_{\geq 0}$, each with finite expectation, such that conditions $(C1^*)$, $(C2)$ and $(C3)$ hold. For any $b > 0$, define $T = \inf\{t \geq 0 \mid X_t \geq b\}$. If $X_0 \in [b]$, then $\mathrm{E}(T) \leq (b^2 - X_0^2)/\delta$. Moreover, for $\tau > 0$, $\Pr(T \geq \tau) \leq e^{-\tau\delta/eb^2}$.*

Theorem 2 tells us that under the standard variance drift case as discussed in [Kötzing, 2016], we can derive an exponential tail bound for the runtime and such a process exhibits a high concentration around the expectation.

Now, we present a corollary which consists of the fixed step size condition.

**Corollary 3.** *Let $(X_t)_{t\geq 0}$ be random variables over $\mathbb{R}_{\geq 0}$, each with finite expectation which satisfy conditions $(C1)$, $(C2)$ and $(C3)$. For any $b > 0$, define $T = \inf\{t \geq 0 \mid X_t \geq b\}$. Given that $X_0 \in [0, b]$, then $\mathrm{E}(T) \leq (b^2 - X_0^2)/\delta$. Moreover, for $\tau > 0$, $\Pr(T \geq \tau) \leq e^{-\tau\delta/eb^2}$.*

Furthermore, we derive a tail bound for the variance-dominated processes with two absorbing states. Following the setting of Theorem 10 in [Göbel *et al.*, 2022], we will prove the next theorem.

**Theorem 4.** *Let $(X_t)_{t\geq 0}$ be random variables over $\mathbb{R}_{\geq 0}$, each with finite expectation such that $(C1*)$, $(C3)$ and $\mathrm{E}(X_{t+1} - X_t \mid \mathcal{F}_t) = 0$ hold. For any $b > 0$, define $T = \inf\{t \geq 0 \mid X_t \in \{0, b\}\}$. If $X_0 \in [0, b]$, then $\mathrm{E}(T) \leq (X_0(b - X_0))/\delta$.*

*Moreover, for $\tau > 0$, $\Pr(T \geq \tau) \leq e^{-2\tau\delta/eb^2}$.*

This proof is similar to Theorem 1 except that we use a different stochastic process $Y_t = X_t(b - X_t) + \delta t$ and show that $Y_t$ is a super-martingale. Unlike the proof of Theorem 1, the proof of Theorem 4 uses the extended Optional Stopping Time Theorem for super-martingale [Williams, 1991]. We defer the proof to the appendix.

### 3.3 Standard Drift

If a stochastic process has drift $\varepsilon$ where $\varepsilon > 0$ is some positive constant, then we can give a different proof for the upper tail bound for additive drift from the proof in [Kötzing, 2016]. This provides a more precise exponential upper tail bound.

**Theorem 5.** *Let $(X_t)_{t\geq 0}$ be random variables over $\mathbb{R}$, each with finite expectation which satisfy condition $(C1*)$ and $\mathbb{E}[X_{t+1} - X_t \mid \mathcal{F}_t] \geq \varepsilon$ for some $\varepsilon > 0$. For any $b > 0$, define $T = \inf\{t \geq 0 \mid X_t \geq b\}$. If $X_0 \in [0, b]$, then $E(T) \leq \frac{b - X_0}{\varepsilon}$. Moreover, for $\tau > 0$, $\Pr(T \geq \tau) \leq e^{-\tau\varepsilon/eb}$.*

We recover the exponential upper tail bound for the additive drift theorem. The bound we obtain gets rid of the coefficients $1/8c^2$ in [Kötzing, 2016]. Theorem 2.5.12 of [Menshikov *et al.*, 2016] provides a similar result and uses a similar recurrence proof idea. While our result generalises the result in [Menshikov *et al.*, 2016] by releasing the fixed step size, we provide a meaningful bound on the first hitting time instead of bounding it above by infinity.

In summary, we have discovered a simple alternative to the Azuma-Hoeffding inequality that provides an exponential tail bound by only relying on basic martingale theory. This result can be applied to the random local search (RLS) type algorithms that make finite steps at each iteration, as well as other randomised algorithms including evolutionary algorithms (EAs) that account for the possibility of large jumps occurring. Another benefit from Theorem 1 is that it allows the tolerance of the negative drift up to $-\Omega(1/b)$.

## 4 Applications to Random 2-SAT and Graph Colouring

In this section, we illustrate our theorems on practical examples. We consider the examples provided by [McDiarmid, 1993; Mitzenmacher and Upfal, 2005; Göbel *et al.*, 2022]. which include variance-dominated processes. We first discuss the Random 2-SAT problem.

### 4.1 Applications to Random 2-SAT

The 2-SAT Algorithm is designed to solve instances of the 2-SAT problem, where a formula consists of clauses and each of them contains exactly two literals (either variables or negations). In each iteration, the algorithm selects an unsatisfied clause and picks one of the literals uniformly at random. The truth value of the variable corresponding to this literal is then inverted. Repeat the process until either we meet the stopping criteria or a valid truth assignment is found.

[Papadimitriou, 1991] firstly provided a time complexity analysis on such a simple randomised algorithm that returns a satisfying assignment of a satisfiable 2-SAT formula $\phi$ with $n$ variables. Later, [Göbel *et al.*, 2022] recovered the results using drift analysis tools which we put in the Appendix.

By applying Theorem 2 with a variance bound 1, we can bound the number of function evaluations of order $O(n^4)$ with an upper exponential tail bound.

**Theorem 6.** *Given any $r \geq 0$, the randomised 2-SAT algorithm, when run on a satisfiable 2-SAT formula over $n \in \mathbb{N}$ variables, terminates in at most $rn^4$ time with probability at least $1 - e^{-r/e}$.*

### 4.2 Applications to Graph Colouring

Now we consider graph colouring, which has already been studied by [McDiarmid, 1993] and [Göbel *et al.*, 2022]. The recolour algorithm generates a 2-colouring mapping with the condition that no monochromatic edges can be found. The algorithm assumes a subroutine called SEEK, which, given a 2-colouring of the points, outputs a monochromatic edge if one exists. If there are no monochromatic edges, then the algorithm terminates. Otherwise, the algorithm repeats picking a point uniformly at random from the given monochromatic edge and changes its colour.

[Göbel *et al.*, 2022] provided a simpler proof of the $O(n^4)$ expected runtime of the recolouring algorithm for finding a 2-colouring with no monochromatic triangles on 3-colorable graphs. Following the setting and the proof of [Göbel *et al.*, 2022], by using Theorem 4, we can derive the following:

**Theorem 7.** *Given any $r \geq 0$, the randomised Recolouring algorithm on a 3-colorable graph with $n \in \mathbb{N}$ vertices over $n \in \mathbb{N}$ variables, terminates in at most $rn^4$ time with probability at least $1 - e^{-4r/3e}$.*

## 5 Applications to Coevolutionary Algorithms

Next, we consider a slightly complicated example: competitive co-evolutionary algorithms (CoEAs). Competitive co-evolutionary algorithms are designed to solve maximin optimisation or adversarial optimisation problems, including two-player zero-sum games [Popovici *et al.*, 2012; Lehre, 2022]. There are various applications, including CoEA-GAN [Toutouh *et al.*, 2019], competitive co-evolutionary search heuristics on cyber security problem [Lehre *et al.*, 2023] and enhanced GANs by using a co-evolutionary approach for image translation [Shu *et al.*, 2019].

We are interested in whether competitive CoEAs can help find Nash equilibrium efficiently. We use the formulation in [Nisan *et al.*, 2007] to define Nash equilibrium. This paper focuses on Pure Strategy Nash Equilibrium (abbreviated NE).

**Definition 4.** ([Nisan *et al.*, 2007]) Consider a two-player zero-sum game. Given a search space $\mathcal{X} \times \mathcal{Y}$ and a payoff function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, if for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$

$$g(x, y^*) \leq g(x^*, y^*) \leq g(x^*, y).$$

then $(x^*, y^*)$ is called a Pure Strategy Nash Equilibrium of a two-player zero-sum game.

The pairwise dominance relation has been defined and introduced into a population-based CoEA in [Lehre, 2022].

**Definition 5** (Pairwise dominance)**.** Given a function $g = \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and two pairs $(x_1, y_1), (x_2, y_2) \in \mathcal{X} \times \mathcal{Y}$, we say that $(x_1, y_1)$ dominates $(x_2, y_2)$ with respect to $g$, denoted $(x_1, y_1) \succeq_g (x_2, y_2)$, if and only if $g(x_1, y_2) \geq g(x_1, y_1) \geq g(x_2, y_1)$.

There is a single-pair CoEA called Randomised Local Search with Pairwise Dominance (RLS-PD) [Hevia Fajardo *et al.*, 2023]. It has been shown that RLS-PD can find the NE of a simple pseudo-Boolean benchmark called BILINEAR in expected polynomial runtime. The processes induced by RLS-PD on the BILINEAR problem is exactly a variance-transformed process. We would like to use Theorem 1 to show the exponential tail bound for the runtime.

RLS-PD samples a search point (a pair of point $(x_1, y_1) \in \mathcal{X} \times \mathcal{Y}$) uniformly at random. In each iteration, RLS-PD uses the local mutation operator to generate the new search point where the local mutation operator is sampling a random Hamming neighbour. If the new search point dominates the original search point in a pairwise-dominance manner, then the original search point is replaced by the new one. Otherwise, the original search point remains the same.

### 5.1 The BILINEAR Problem

In this section, we consider a simple class of discrete maximin benchmark called BILINEAR, which was first proposed by [Lehre, 2022]. In this work, we use a variation of BILINEAR as [Hevia Fajardo *et al.*, 2023] to simplify our calculation and illustrate applications of our main theorem. It has been empirically shown in [Hevia Fajardo *et al.*, 2023] that RLS-PD behaves similarly on the original definition of BILINEAR and the revised definition. In this paper, we only consider this variation of BILINEAR.

**Definition 6.** ([Hevia Fajardo *et al.*, 2023]) The BILINEAR function is defined for two parameters $\alpha, \beta \in (0, 1)$ by

$$\textsc{Bilinear}_{\alpha,\beta}(x, y) := |y|_1 (|x|_1 - \beta n) - \alpha n |x|_1 + E_1 + E_2$$

with the error terms $E_1 := \max\{(\alpha n - |y|_1)^2, 1\}/n^3$ and $E_2 := -\max\{(\beta n - |x|_1)^2, 1\}/n^3$. We also denote the set of Nash equilibria as OPT, where $\mathrm{OPT} := \{(x, y) \mid |x|_1 = \beta n \wedge |y|_1 = \alpha n\}$.

We consider OPT as our solution concept and the problem setting $\alpha = 1/2 \pm O(1/\sqrt{n})$ and $\beta = 1/2 \pm O(1/\sqrt{n})$ as in [Hevia Fajardo *et al.*, 2023]. We now derive the exponential tail bound of RLS-PD to find the Nash equilibrium.

### 5.2 RLS-PD Solves BILINEAR Efficiently W.H.P.

**Theorem 8.** *Consider $\alpha \in [1/2 - A/\sqrt{n}, 1/2 + A/\sqrt{n}]$ and $\beta \in [1/2 - B/\sqrt{n}, 1/2 + B/\sqrt{n}]$, where $A, B > 0$ are constants and $3(A + B)^2 \leq 1/2 - \delta'$ for some constant $\delta' > 0$. The expected runtime of RLS-PD on $\textsc{Bilinear}_{\alpha,\beta}$ is $O(n^{1.5})$. Moreover, given any $r \geq 0$, the runtime is at most $2rn^{1.5}$, with probability at least $1 - e^{-\Omega(r)}$.*

We defer the proof of Theorem 8 to the appendix.

Theorem 8 shows that RLS-PD can find the Nash Equilibrium in $O(n^{1.5})$ with overwhelmingly high probability. The exponential tail bound provides a stronger performance guarantee up to the tail of the runtime than the sole expectation.

### 5.3 RLS-PD Forgets the Nash Equilibrium W.H.P.

After the algorithm finds a Nash Equilibrium efficiently, the inherent characteristics of the function cause the algorithm not only to forget the Nash Equilibrium but also move away from it by a distance $\Omega(\sqrt{n})$ in $O(n)$ iterations w.h.p. This is shown by the following theorem.

**Theorem 9.** *Let $\alpha = 1/2 \pm A/\sqrt{n}$ and $\beta = 1/2 \pm B/\sqrt{n}$, where $A, B > 0$ are constants. Consider RLS-PD on $\textsc{Bilinear}_{\alpha,\beta}$. Then, for any initial search points $(x_0, y_0)$, the expected runtime that the search point firstly moves away from OPT by a Manhattan distance at least $(A + B)\sqrt{n}$ is $O(n)$. Moreover, given any $r > 0$, the runtime is at most $rn$, with probability at least $1 - e^{-\Omega(r)}$.*

Theorem 9 illustrates how drift analysis can expose weaknesses in algorithms, suggesting what needs to be improved in new algorithms. In particular, we can see even though RLS-PD can find the optimum in polynomial time, it can still suffer from evolutionary forgetting (i.e. forget the optimum found in previous iterations) with high probability. So only the expected runtime estimate might be insufficient to determine whether a coevolutionary algorithm is good or not. This highlights the weakness of RLS-PD and the need to understand coevolutionary dynamics further.

## 6 Applications to Regret Analysis of a Bandit Learning Algorithm

We start with a brief introduction of bandit problems. Suppose we have $K$ decisions or "arms", where we obtain the corresponding reward $r_a$ when we choose one specific decision $a$. The goal of the bandit algorithm is to maximise cumulative reward among time horizon $T$ [Lattimore and Szepesvári, 2020; Sutton and Barto, 2018]. In this paper, we consider the quantity called regret (missed reward), which is the difference between the reward of the chosen arm and the optimal arm at each iteration. We provide a formal definition of regret as follows.

**Definition 7.** ([Lattimore and Szepesvári, 2020; Larcher *et al.*, 2023]) Given time horizon $T$, each arm $a$ is associated with a probability distribution $\mathcal{D}(a)$, which we assume to be over $[0, 1]$ and for which the mean is denoted as $\mu(a)$; whenever arm $a$ is pulled, the agent receives a reward distributed according to $\mathcal{D}(a)$. The regret (missed reward) of the agent at round $t \in \mathbb{N}$ is defined as $R_t = r_{a^*} - r_{a_t}$, where $a_t$ is the arm chosen at round $t$, $r_a$ is the reward obtained from reward distribution $\mathcal{D}(a)$ and $a^* = \arg\max\{\mu(a) \mid a \in [K]\}$. The goal of the agent is to minimise the total regret $\mathcal{R} = \sum_{t=1}^{T} R_t$ or the total expected regret $\mathrm{E}(\mathcal{R})$.

This paper focuses on the non-stationary 2-armed bandit problem, in which the reward distributions may swap over time and $K = 2$. More precisely, the agent receives a reward according to a reward distribution $\mathcal{D}(a_1)$ by pulling arm $a_1$ and another reward according to another reward distribution $\mathcal{D}(a_2)$ by pulling arm $a_2$. We assume that two distributions

$\mathcal{D}(a_1)$ and $\mathcal{D}(a_2)$ are fixed, and they will swap if a change occurs along the time horizon. To simplify the calculation, we assume both distributions over $[0,1]$. We present the application of our drift theorem (concentration tail bound) on regret analysis of a simple reinforcement learning algorithm for such a bandit problem. We defer the pseudo-code for Random Walk with Asymmetric Boundaries (RWAB) proposed by [Larcher *et al.*, 2023] to the appendix.

Note that RWAB is designed to balance the exploration and exploitation for non-stationary bandit problems. RWAB mainly relies on the CHALLENGE operator to determine which arm we prefer to pull or whether we swap the arms. The CHALLENGE operator is designed to use the random walk of action value $S$ on $[-\sqrt{T/L}, 1]$. [Larcher *et al.*, 2023] shows that the expected regret of RWAB is $\Theta\left(\sqrt{LT}\right)$ where $T$ is the time horizon and $L$ is the number of changes. We want stronger performance guarantees for the regret of RWAB, i.e. a concentration tail bound for the regret estimate. We would like to characterise the distribution of the regret.

Next, we present the main theorem for the regret of RWAB algorithm. In this theorem, we assume $L = o(T)$.

**Theorem 10.** *Given any* $\varepsilon \geq 1$*, the regret of Algorithm* RWAB *is at most* $480\varepsilon(L + \sqrt{LT})$ *with probability at least* $1 - 2e^{-\sqrt{\varepsilon}/e}$.

The proof of Theorem 10 is deferred to the Appendix.

By Theorem 10, RWAB Algorithm has regret at most order $O\left(\sqrt{LT}\right)$ for a 2-armed non-stationary bandit problem w.h.p. By using minimax lower bound [Bubeck *et al.*, 2012], any algorithm on $K$-armed stationary bandit problems has regret at least $\sqrt{Kn}/20$ for time horizon $n$. In particular, for $K = 2$, the lower bound for the expected regret of any algorithm on bandit stationary bandit problems is $\Omega(\sqrt{n})$ for time horizon $n$. [Larcher *et al.*, 2023] showed that by considering $L$ changes and each change of average steps $T/L$, RWAB has expected regret at least $\Omega\left(L \cdot \sqrt{T/L}\right) = \Omega\left(\sqrt{LT}\right)$. Theorem 10 confirms that RWAB is optimal with overwhelmingly high probability, and we propose a new perspective of analysing a bandit algorithm by using drift analysis, which is rarely employed by the reinforcement learning community.

## 7 Experiments

To complement our asymptotic results with data for concrete problem sizes, we conduct the following experiments.

### 7.1 Empirical Evidence of RLS-PD on BILINEAR

We conduct experiments with the RLS-PD for the maximin BILINEAR problem. The problem setup is $(\alpha, \beta) = (0.5, 0.5), (0.3, 0.3), (0.3, 0.7), (0.7, 0.3), (0.7, 0.7)$. These five scenarios cover the cases when the optimum lies in four different quadrants and the centre of the search space. We set the mutation rate $\chi = 1$ and problem size $n = 1000$. We run 1000 independent simulations for each configuration. For each run, we initialise the search point uniformly at random.

Figure 1 displays the density plot for the runtime distribution of RLS-PD on BILINEAR. The $x$-axis represents the runtime, the $y$-axis represents the frequency or density, and the red dotted line represents the average value of the runtime for each problem setting. As $x$ increases, Figure 1 shows that we have an exponentially decaying tail for the runtime of RLS-PD on each problem configuration. It is very unlikely that the runtime of RLS-PD on BILINEAR deviates too much from the mean or the expected runtime from Figure 1 for each problem configuration. From the statistics (figures and tables in the appendix), we can see for each configuration, the frequency that the actual runtime bounded above by the mean runtime converges to 1. When $(\alpha, \beta) = (0.5, 0.5)$, the empirical results are consistent with our theoretical bounds in the sense of asymptotic order. The results for other problem configurations raise a conjecture about whether our theoretical results can also hold for all $\alpha, \beta \in [0, 1]$.

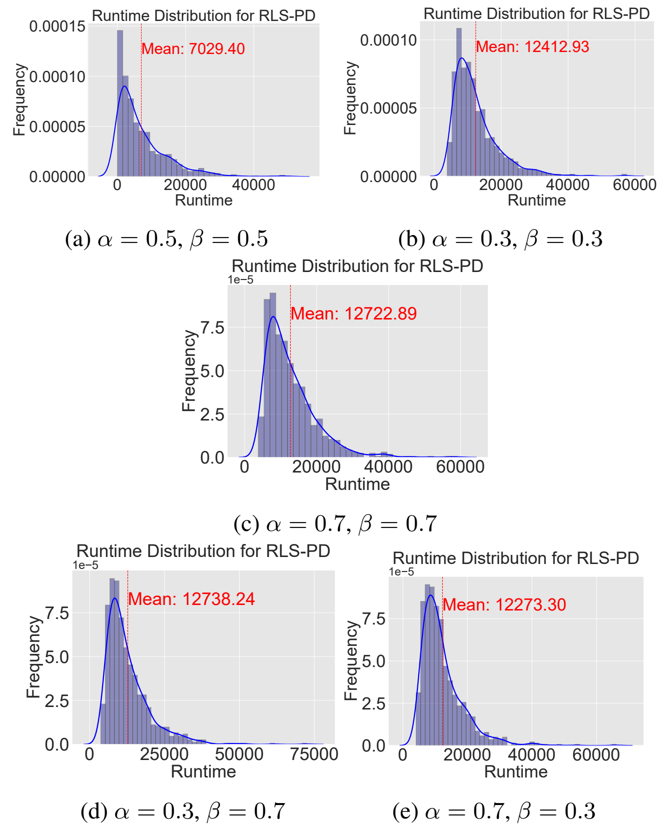
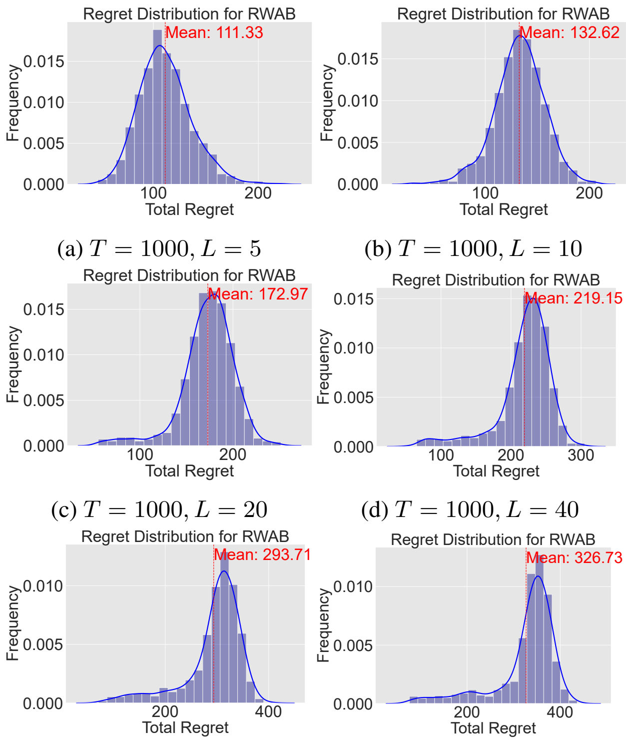



Figure 1: Runtime distribution for RLS-PD for various $\alpha$ and $\beta$.

### 7.2 Empirical Evidence of RWAB Algorithm

We conduct experiments with the RWAB Algorithm for the 2-armed non-stationary bandit problem. The environment is set up as two Bernoulli bandits with means $\mu_1 = 0.2, \mu_2 = 0.8$ and the number of changes $L = 5, 10, 20, 40, 80, 100$. The changes are set up uniformly at random along the time horizon $T = 1000$. 1000 independent simulations are run for each configuration.

Figure 2 displays the regret distribution of RWAB. The $x$-axis represents the regret of RWAB, the $y$-axis represents the frequency or density, and the red dotted line represents the




(a) $T = 1000, L = 5$ (b) $T = 1000, L = 10$ (c) $T = 1000, L = 20$ (d) $T = 1000, L = 40$ (e) $T = 1000, L = 80$ (f) $T = 1000, L = 100$

Figure 2: Regret distribution for various values of $T$ and $L$.

average regret for each problem setting. As $x$ increases, Figure 2 shows that we have an exponentially decaying tail for the regret of RWAB on 2-armed non-stationary bandit problem with respect to the changes $L$. Figure 2 shows the concentration of regret around the empirical mean or the expected regret and it is also unlikely that the regret of RWAB deviates too much from the expectation. The tables (deferred in the appendix) suggest that our theoretical tail bound is asymptotically tight regardless of the leading coefficient. Tables and figures for regret distributions show that for each configuration, the frequency that the actual regret bounded above by the mean regret is asymptotic to 1 as the increases in the multiplicative factors of the upper bounds. Moreover, the convergence rate is significantly faster than the counterpart in the case of RLS-PD on BILINEAR. This means that the theoretical bound (i.e. the leading coefficient) obtained has room to improve. One conjecture may be the process governing the dynamics of runtime for RLS-PD on BILINEAR relies heavily on the high variance needed to overcome the negative drift, while the process governing the dynamics of regret induced by RWAB already exhibits positive drift everywhere before reaching the target state. Thus, it yields a faster convergence.

## 8 Conclusion

This paper proves a more general and stronger drift theorem (tail-bound). Our theorems can be used to analyse the first hitting time of different random processes. As a sub-product, this paper also resolves the open problem left in [Kötzing, 2016], which asks for a suitable replacement for the Azuma-Hoeffding inequality to improve the tail bounds for random processes. We apply our theorems to several practical examples, including Random 2-SAT, Recolouring, competitive CoEAs and RWAB. To the best of our knowledge, it is the first tail-bound drift analysis of RLS-PD and RWAB. Our drift theorems provide more precise information on how the runtime concentrates and a stronger performance guarantee. In practice, it shows the limitation of the current coevolutionary algorithm on maximin optimisation. It suggests a need for a deeper understanding of the mechanism of CoEAs, which may help to design a more stable CoEA. Moreover, our results confirm that randomness in RWAB can be helpful for stochastic non-stationary bandit problems.

For future studies, both runtime analysis of CoEA on maximin optimisation and regret analysis of stochastic reinforcement learning algorithms via drift analysis are still poorly understood and unexplored areas. In particular, on the technical side, can we derive more precise bounds for RWAB since the leading coefficient seems not to be optimal from empirical results or can we use these results to analyse more complicated CoEAs or bandit algorithms? On the practical side, we could try to use such concentration bound to design more efficient algorithms. For example, we could try to design more stable CoEAs or develop a general optimal bandit algorithm by using the random-walk design analysed in this work.

## Acknowledgments

This work was supported by a Turing AI Fellowship (EPSRC grant ref EP/V025562/1). The computations were performed using the University of Birmingham's BlueBEAR high performance computing (HPC) service.

## Contribution Statement

Authors are listed in alphabetical order.

## References

[Popovici *et al.*, 2012] Elena Popovici, Anthony Bucci, R. Paul Wiegand, and Edwin D. De Jong. Coevolutionary Principles. In Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok, editors, *Handbook of Natural Computing*, pages 987–1033. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Al-Dujaili *et al.*, 2018] Abdullah Al-Dujaili, Tom Schmiedlechner, and Erik Hemberg, and Una-May O'Reilly. Towards Distributed Coevolutionary GANs, August 2018. arXiv:1807.08194 [cs].

[Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

[Azuma, 1967] Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19:357–367, 1967.

[Baritompa and Steel, 1996] Bill Baritompa and Mike Steel. Bounds on absorption times of directionally biased random sequences. *Random Structures & Algorithms*, 9(3):279–293, 1996.

[Bhattacharya and Waymire, 2007] Rabindra Nath Bhattacharya and Edward C Waymire. *A Basic Course in Probability Theory*, volume 69. Springer, 2007.

[Bian *et al.*, 2020] Chao Bian, Chao Feng, Chao Qian, and Yang Yu. An efficient evolutionary algorithm for subset selection with general cost constraints. In *AAAI*, 2020.

[Bubeck *et al.*, 2012] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[Case and Lehre, 2020] Brendan Case and Per Kristian Lehre. Self-adaptation in nonelitist evolutionary algorithms on discrete problems with unknown structure. *IEEE Transactions on Evolutionary Computation*, 24(4):650–663, 2020.

[Dang *et al.*, 2021] Duc-Cuong Dang, Anton Eremeev, and Per Kristian Lehre. Escaping local optima with non-elitist evolutionary algorithms. In *AAAI*, 2021.

[Doerr and Goldberg, 2013] Benjamin Doerr and Leslie Ann Goldberg. Adaptive Drift Analysis. *Algorithmica*, 65(1):224–250, January 2013.

[Doerr and Künnemann, 2013] Benjamin Doerr and Marvin Künnemann. How the (1+$\lambda$) evolutionary algorithm optimizes linear functions. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, GECCO '13, page 1589–1596, New York, NY, USA, 2013. Association for Computing Machinery.

[Doerr and Neumann, 2019] Benjamin Doerr and Frank Neumann. *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*. Natural Computing Series. Springer Nature, 2019.

[Doerr and Qu, 2023] Benjamin Doerr and Zhongdi Qu. Runtime analysis for the nsga-ii: Provable speed-ups from crossover. In *AAAI*, 2023.

[Doerr and Zheng, 2020] Benjamin Doerr and Weijie Zheng. Sharp bounds for genetic drift in estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 24(6):1140–1149, 2020.

[Doerr *et al.*, 2010] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 1449–1456, 2010.

[Doerr *et al.*, 2013] Benjamin Doerr, Dirk Styled, and Carsten Witt. When do evolutionary algorithms optimize separable functions in parallel? In *Proceedings of the Twelfth Workshop on Foundations of Genetic Algorithms XII*, FOGA XII '13, page 51–64, New York, NY, USA, 2013. Association for Computing Machinery.

[Doob, 1971] Joseph Leo Doob. What is a Martingale? *The American Mathematical Monthly*, 78(5):451–463, 1971. Publisher: Mathematical Association of America.

[Friedrich *et al.*, 2016] Tobias Friedrich, Timo Kötzing, and Martin S. Krejca. Edas cannot be balanced and stable. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, page 1139–1146, New York, NY, USA, 2016. Association for Computing Machinery.

[Göbel *et al.*, 2022] Andreas Göbel, Timo Kötzing, and Martin S Krejca. Intuitive analyses via drift theory, 2022.

[Grimmett and Stirzaker, 2001] Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford University Press, Oxford ; New York, 3rd ed edition, 2001.

[Hajek, 1982] Bruce Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied probability*, 14(3):502–525, 1982.

[He and Yao, 2001] Jun He and Xin Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127(1):57–85, March 2001.

[Hevia Fajardo *et al.*, 2023] Mario Alejandro Hevia Fajardo, Per Kristian Lehre, and Shishen Lin. Runtime analysis of a co-evolutionary algorithm: Overcoming negative drift in maximin-optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '23, New York, NY, USA, 2023. Association for Computing Machinery.

[Jansen, 2013] Thomas Jansen. *Analyzing Evolutionary Algorithms: The Computer Science Perspective*. Springer Publishing Company, Incorporated, 2013.

[Johannsen, 2010] Daniel Johannsen. *Random combinatorial structures and randomized search heuristics*. PhD thesis, Saarland University, 2010.

[Kötzing *et al.*, 2015] Timo Kötzing, Andrei Lissovoi, and Carsten Witt. (1+1) ea on generalized dynamic onemax. In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII*, FOGA '15, page 40–51, New York, NY, USA, 2015. Association for Computing Machinery.

[Kötzing, 2016] Timo Kötzing. Concentration of First Hitting Times Under Additive Drift. *Algorithmica*, 75(3):490–506, July 2016.

[Larcher *et al.*, 2023] Maxime Larcher, Robert Meier, and Angelika Steger. A simple optimal algorithm for the 2-arm bandit problem. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 365–372. SIAM, 2023.

[Lattimore and Szepesvári, 2020] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[Lehre and Lin, 2023] Per Kristian Lehre and Shishen Lin. Is CC-(1+1) EA more efficient than (1+1) EA on separable and inseparable problems? In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, Chicago, USA, 2023.

[Lehre and Witt, 2021] Per Kristian Lehre and Carsten Witt. Tail bounds on hitting times of randomized search heuristics using variable drift analysis. *Combinatorics, Probability and Computing*, 30(4):550–569, 2021.

[Lehre *et al.*, 2023] Per Kristian Lehre, Mario Alejandro Hevia Fajardo, Jamal Toutouh, Erik Hemberg, and Una-May O'Reilly. Analysis of a pairwise dominance coevolutionary algorithm and defendit. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, page 1027–1035, New York, NY, USA, 2023. Association for Computing Machinery.

[Lehre, 2022] Per Kristian Lehre. Runtime Analysis of Competitive co-Evolutionary Algorithms for Maximin Optimisation of a Bilinear Function. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, page 1408–1416, 2022.

[McDiarmid, 1989] Colin McDiarmid. *On the method of bounded differences*, page 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press, 1989.

[McDiarmid, 1993] Colin McDiarmid. A random recolouring method for graphs and hypergraphs. *Combinatorics, Probability and Computing*, 2(3):363–365, 1993.

[Menshikov *et al.*, 2016] Mikhail Menshikov, Serguei Popov, and Andrew Wade. *Non-homogeneous Random Walks: Lyapunov Function Methods for Near-Critical Stochastic Systems*. Cambridge Tracts in Mathematics. Cambridge University Press, 2016.

[Mitavskiy *et al.*, 2009] Boris Mitavskiy, Jonathan Rowe, and Chris Cannings. Theoretical analysis of local search strategies to optimize network communication subject to preserving the total number of links. *International Journal of Intelligent Computing and Cybernetics*, 2(2):243–284, 2009.

[Mitzenmacher and Upfal, 2005] Michael Mitzenmacher and Eli Upfal. *Probability and computing: an introduction to randomized algorithms and probabilistic analysis*. Cambridge University Press, New York, 2005.

[Neumann and Witt, 2010] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*. Natural Computing Series. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[Nisan *et al.*, 2007] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vazirani Vijay V. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[Oliveto and Witt, 2012] Pietro S. Oliveto and Carsten Witt. Erratum: Simplified Drift Analysis for Proving Lower Bounds in Evolutionary Computation, November 2012. arXiv:1211.7184 [cs].

[Papadimitriou, 1991] Christos H Papadimitriou. On selecting a satisfying truth assignment. In *Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 163–169. IEEE Computer Society, 1991.

[Shu *et al.*, 2019] Han Shu, Yunhe Wang, Xu Jia, Kai Han, Hanting Chen, Chunjing Xu, Qi Tian, and Chang Xu. Co-evolutionary compression for unpaired image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3235–3244, 2019.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Williams, 1991] David Williams. *Probability with martingales*. Cambridge university press, 1991.

[Zheng *et al.*, 2022] Weijie Zheng, Yufei Liu, and Benjamin Doerr. A first mathematical runtime analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In *AAAI*, 2022.

[Toutouh *et al.*, 2019] Jamal Toutouh, Erik Hemberg, and Una-May O'Reilly. Spatial Evolutionary Generative Adversarial Networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 472–480. ACM, Prague Czech Republic, July 2019.