Towards Runtime Analysis of Population-Based Co-evolutionary Algorithms on Sparse Binary Zero-Sum Games

Per Kristian Lehre, Shishen Lin*

School of Computer Science, University of Birmingham, Birmingham, United Kingdom
p.k.lehre@bham.ac.uk, sxl1242@student.bham.ac.uk

Abstract

The maximin optimisation problem, inspired by Von Neumann's work (von Neumann 1928) and widely applied in adversarial optimisation, has become a key research area in machine learning. Gradient Descent Ascent (GDA) is a common method for solving these problems but requires the payoff function to be differentiable, making it unsuitable for discrete or binary functions that often occur in game-theoretical scenarios. Co-evolutionary algorithms (CoEAs), which are derivative-free, offer an alternative to these problems. However, the theoretical understanding of CoEAs is still limited.

This paper provides the first rigorous runtime analysis of CoEAs with pairwise dominance on binary two-player zero-sum games (or maximin problems), specifically focusing on the DIAGONAL game. The mathematical analysis rigorously shows that the PDCoEA can efficiently find the optimum in polynomial runtime with high probability under low mutation rates and large population sizes. Empirical evidence also identifies an error threshold where higher mutation rates lead to inefficiency. In contrast, single-pair-individual algorithms, i.e., RLS-PD and (1+1)-CoEAs, fail to find the optimum in polynomial time. These findings highlight the usefulness of pairwise dominance, low mutation rates, and large populations in maintaining a "co-evolutionary arms race".

Introduction

The maximin problem is a significant class of optimisation problems in machine learning, with applications in adversarial optimisation, training, game playing, and matrix games (Lin, Jin, and Jordan 2020; Palaniappan and Bach 2016; Mertikopoulos et al. 2018; Daskalakis and Panageas 2018; Goodfellow et al. 2014; Robey et al. 2024; Zhang et al. 2022; Gidel, Jebara, and Lacoste-Julien 2017; Grand-Clément and Kroer 2021; Wei et al. 2021). When gradient information is available, gradient descent ascent (GDA) is a natural approach, especially for bilinear objectives like $f(x, y) = x^T Ay$ (Gidel, Jebara, and Lacoste-Julien 2017; Daskalakis and Panageas 2018). However, when gradients are inaccessible but the problem is convex-concave, zeroth-order methods can provide solutions (Dvinskikh et al. 2022; Beznosikov, Sadiev, and Gasnikov 2020; Gasnikov et al. 2022). The absence of gradients or a nonconvex-nonconcave structure, however, makes these problems intrinsically more challenging (Colson, Marcotte, and Savard 2007).

Evolutionary Algorithms (EAs) are biologically inspired randomised heuristics designed to find global optima with minimal knowledge of fitness functions, making them ideal for black-box or oracle settings, especially where derivative-free methods are needed (Popovici et al. 2012; Eiben and Smith 2015; Xue et al. 2021). Co-evolutionary algorithms (CoEAs), a subset of EAs, are particularly promising for tackling nonconvex-nonconcave and non-differentiable optimisation problems within cooperative or non-cooperative game-theoretical scenarios. These methods have been applied in areas like coevolutionary learning, defence, design of sorting network and novelty search (Xue et al. 2023; Hemberg et al. 2021; Hevia Fajardo et al. 2024; Benford et al. 2024; Ficici and Pollack 1998; Hillis 1990; O'Hanley and Church 2011; Gomes, Mariano, and Christensen 2014; Wang et al. 2019, 2020). However, a significant research gap remains in the theoretical understanding of coevolution.

Runtime analysis studies the number of iterations or function evaluations algorithms take before finding the optimum. Runtime analysis not only provides more precise performance guarantees and guidelines for practitioners to design randomised algorithms but also enhances the explainability of various heuristic search and black-box optimisation algorithms, including NSGA-II (Non-dominated Sorting Genetic Algorithm II), MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition) and EDAs (Estimation-of-Distribution Algorithms) (Do et al. 2023; Zheng and Doerr 2023; Doerr and Qu 2023; Zheng, Liu, and Doerr 2022).

Related Work: Theoretical analysis of evolutionary algorithms on discrete maximin problems is limited. Recent studies (Lehre 2022; Hevia Fajardo, Lehre, and Lin 2023) have conducted rigorous runtime analyses of CoEAs on discrete maximin problems such as DISCRETE-BILINEAR, demonstrating that single-pair CoEAs and population-based CoEAs with pairwise dominance can solve DISCRETE-BILINEAR in polynomial runtime. However, there is a gap in the theoretical understanding of CoEAs on more challenging discrete games with binary outputs (i.e. $g : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$), as originated from seminal works on co-evolving sorting networks and test cases by Hillis (1990) and co-evolving players in the game of

*Authors are listed in alphabetical order.

Backgammon by Pollack and Blair (1998). Lehre and Lin (2024) showed that $(1,\lambda)$-CoEA can find the approximation of the optimum of the DIAGONAL game in polynomial runtime with high probability. Moreover, another variant of DIAGONAL called the bigger-number game was considered in (Zhang and Sandholm 2024) to show the inefficiency of the Double Oracle Algorithm in zero-sum games. However, the analytic tool developed in previous related works cannot be used directly in our scenario due to the complexity of the binary games and the difficulty of analysing the population-based CoEAs. Thus, we adapt the level-based theorem for co-evolution to study this problem (see Theorem 1).

Our Contributions: This paper tackles the open challenges in zero-sum games with sparse binary payoff signals, presenting the first rigorous runtime analysis of CoEAs with pairwise dominance on DIAGONAL game. Specifically, we demonstrate that single-pair CoEAs, such as Random Local Search with Pairwise Dominance (RLS-PD) and (1+1)-CoEA, cannot find the optimum of DIAGONAL in polynomial runtime with high probability. Additionally, we show that a population-based CoEA (PDCoEA) can find the optimum in polynomial runtime under large population sizes and low constant mutation rates. Our experimental results further illustrate the practical implications of our theoretical bounds and examine the error threshold for mutation rates in PDCoEA. Moreover, these findings highlight the promising potential of coevolution for binary maximin problems and reveal the importance of low mutation rates and large population sizes in maintaining a "coevolutionary arms race".

Preliminaries & Problem

Notations: For a filtration $\mathcal{F}_t$, we write $\mathrm{E}_t(\cdot) := \mathrm{E}(\cdot|\mathcal{F}_t)$ and $\Pr_t(\cdot) := \Pr(\cdot|\mathcal{F}_t)$. We denote 1-norm by $|z|_1 = \sum_{i=1}^n z_i$ for $z \in \{0,1\}^n$. $[n] := \{1, 2, \ldots, n\}$. "With high probability" is abbreviated to "w.h.p.". $o(1)$ of any function $f(n)$ where $n \in \mathbb{N}$ means that $\lim_{n\to\infty} f(n) = 0$. $x \in \mathrm{poly}(n)$ denotes that there exist constants c, d such that $x \le cn^d$. A two-player game is defined by the strategy spaces $\mathcal{X}$ and $\mathcal{Y}$, along with payoff functions $g_i : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $i \in [2]$. Here, $g_i(x,y)$ represents the payoff to player i when player 1 adopts strategy x and player 2 adopts strategy y. NE denotes the Nash Equilibrium.

In this paper, we denote $X_t = |x_t|_1 \in [n] \cup \{0\}$ for $x_t \in \{0,1\}^n$ and $Y_t = |y_t|_1 \in [n] \cup \{0\}$ for $y_t \in \{0,1\}^n$ for any $n \in \mathbb{N}$. We consider the search space $\mathcal{X} \times \mathcal{Y} = \{0,1\}^n \times \{0,1\}^n$ for any $n \in \mathbb{N}$. We consider the filtration $(\mathcal{F}_t)_{t\ge0}$ including the σ-algebra of $(X_0, Y_0), \ldots, (X_t, Y_t)$.

Binary Zero-Sum Games: Given a two-player game with strategy spaces $\mathcal{X}$ and $\mathcal{Y}$, the payoff functions $g_1, g_2 : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ are defined for player 1 and player 2, respectively. The game is zero-sum if player 1's gain is exactly equal to player 2's loss (and vice versa), meaning $g_1(x,y) + g_2(x,y) = 0$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. If $g_1(x,y), g_2(x,y) \in \{-1,1\}$ for all $x, y \in \mathcal{X}$, the game is called binary. Given a binary function $g : \mathcal{X} \times \mathcal{Y} \to \{-1,1\}$, we refer to the game with payoff functions $g_1(x,y) = g(x,y)$ and $g_2(x,y) = -g(x,y)$ as the *binary zero-sum game* defined by g.

Many classical games where the outcomes are win/lose such as Nim, Backgammon, Chess or Go (without draw) can be represented by a binary function g by setting $g(x,y) = 1$ if and only if player 1 wins and $g(x,y) = 0$ otherwise.

Sparsity: This paper adopts the notion of sparsity from (Auger et al. 2015). Note that this is a different notion than games with sparse payoff matrix. Instead, we consider the sparsity in the sense that the support of mixed strategies (non-zero probability terms) is far fewer than the size of pure strategies each player can take.

DIAGONAL Games: To model the binary maximin optimisation problem originating from Hillis's seminal work on co-evolving sorting networks and test cases (Hillis 1990), Lehre and Lin (2024) introduced a class of payoff functions with $\mathcal{X} \times \mathcal{Y}$ as the input and $\{-1,1\}$ as the output, namely DIAGONAL. Here, $\mathcal{X} = \{0,1\}^n$ represents the set of genotypes corresponding to designs for sorting networks, while $\mathcal{Y} = \{0,1\}^n$ denotes the set of genotypes for test cases. $g(x,y) = 1$ if and only if design x passes test case y. The optimisation problem is to find $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ such that

$$\text{for all } (x,y) \in \mathcal{X} \times \mathcal{Y},\ g(x,y^*) \le g(x^*,y^*) \le g(x^*,y).$$

Sparse binary games like DIAGONAL game and its variant have been studied in the literature (Auger et al. 2015; Lehre and Lin 2024; Zhang and Sandholm 2024; Benford and Lehre 2024).

Definition 1 (DIAGONAL Game (Lehre and Lin 2024))**.** Given $n \in \mathbb{N}$, $\mathcal{X} = \mathcal{Y} = \{0,1\}^n$, the payoff function DIAGONAL : $\mathcal{X} \times \mathcal{Y} \to \{-1,1\}$ is

$$\text{DIAGONAL}(x,y) := \begin{cases} 1 & |y|_1 \le |x|_1 \\ -1 & \text{otherwise} \end{cases}.$$

For notational brevity, we will denote the function DIAGONAL by g in the remaining of this paper. A payoff of 1 indicates that design x passes the test cases y; otherwise, the payoff is -1. In the DIAGONAL game, $y^* = 1^n$ is the hardest test case, which only the solution $x^* = 1^n$ can pass (i.e., $g(x^*, y^*) = 1$). Thus, (x^*, y^*) is the unique maximin optimum, where neither the design nor the test case deviates without affecting their payoff $g(x,y)$. This aligns with the NE in zero-sum games. The paper investigates whether CoEAs can efficiently find the NE of DIAGONAL.

Remark. To distinguish DIAGONAL from the commonly used ONEMAX benchmark in the theory of evolutionary computation (Doerr and Neumann 2019), we highlight key differences. Unlike ONEMAX, DIAGONAL is more challenging. ONEMAX uses a linear fitness function (payoff independent of the opponent), while DIAGONAL is a game where payoff depends on the opponent. DIAGONAL provides only 1 bit of information per evaluation, compared to $\log(n)$ bits in ONEMAX. Additionally, ONEMAX has an ascending fitness landscape, making it easy for algorithms like (1+1)-EA or RLS to follow, whereas the binary payoff landscape in DIAGONAL leads to genetic drift, rendering single-pair CoEAs inefficient. Understanding these differences and their impact is a key contribution of this paper.

Co-evolutionary Algorithms

A broad class of co-evolutionary processes, defined by Algorithm 1 update two populations P, Q. At each iteration, a

Algorithm 1: Co-evolutionary Process (Lehre 2022)

Require: Population size $\lambda \in \mathbb{N}$, strategy spaces $\mathcal{X}$ and $\mathcal{Y}$.
Require: Initial populations $P_0 \in \mathcal{X}^\lambda$ and $Q_0 \in \mathcal{Y}^\lambda$.
for each generation number $t \in \mathbb{N}_0$ **do**
  for each interaction number $i \in [\lambda]$ **do**
    Sample an interaction $(x, y) \sim \mathcal{D}(P_t, Q_t)$.
    Set $P_{t+1}(i) := x$ and $Q_{t+1}(i) := y$.

new pair of solutions is sampled from P, Q via the interaction of two populations. Such an interaction includes variation, evaluation, selection, etc. The population is then updated with this new pair of solutions, and the process continues until the termination criteria are met. For single-pair CoEAs, we denote the current pair in generation t by (x_t, y_t). These stochastic processes track the dynamics of coevolutionary algorithms, and we will analyse their first hitting time on the target set. Next, we introduce the concept of pairwise dominance, which is key to characterising the operator $\mathcal{D}$ of CoEAs.

Characteristic Lemma of Dominance Relation

Lehre (2022) demonstrated that a population-based CoEA called Pairwise Dominance CoEA (PDCoEA) can solve some instances of BILINEAR (a bi-linear payoff function with real-valued rather than binary payoffs) in polynomial runtime with high probability. Hevia Fajardo, Lehre, and Lin (2023) shows random local search with pairwise dominance (RLS-PD) solves some instances of BILINEAR in expected polynomial runtime (as summarised by the Table in the Appendix). Therefore, we aim to explore whether such a pairwise dominance method can also work for our binary maximin optimisation problem. Here, we provide a definition of the pairwise dominance relation (c.f. Definition 2).

Definition 2. (Lehre 2022) Given a function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and two pairs $(x_1, y_1), (x_2, y_2) \in \mathcal{X} \times \mathcal{Y}$, (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 $g(x_1, y_2) \geq g(x_1, y_1) \geq g(x_2, y_1)$.

In order to understand how the search point will move under the pairwise dominance relation, we characterise the relationship among search points in DIAGONAL games.

Definition 3. Given a pair of search point $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we say (x, y) lies above the diagonal if $|x|_1 < |y|_1$ and (x, y) lies below the diagonal if $|x|_1 \geq |y|_1$.

Next, we derive sufficient and necessary conditions for the dominance relation of DIAGONAL games. We defer all the proofs to the appendix.

Lemma 1. Given a DIAGONAL game denoted by g, $(x, y) \succeq_g (u, v)$ if and only if $(x, y), (u, v)$ satisfy one of the following:

(1) $|x|_1 < |v|_1, |x|_1 < |y|_1$ and $|u|_1 < |y|_1$;
(2) $|x|_1 \geq |v|_1, |x|_1 < |y|_1$ and $|u|_1 < |y|_1$;
(3) $|x|_1 \geq |v|_1, |x|_1 \geq |y|_1$ and $|u|_1 < |y|_1$;
(4) $|x|_1 \geq |v|_1, |x|_1 \geq |y|_1$ and $|u|_1 \geq |y|_1$.

A Refined Level-Based Theorem

To analyse the runtime of population-based evolutionary algorithms, we define the runtime properly.

Definition 4. For any instance A of Algorithm 1 and $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y}$, define $T_{A,\mathcal{S}} := \min\{t\lambda \in \mathbb{N} \mid (P_t \times Q_t) \cap \mathcal{S} \neq \emptyset\}$.

Although Lehre (2022); Hevia Fajardo, Lehre, and Lin (2023) have analysed CoEAs on BILINEAR, as mentioned in the introduction, DIAGONAL and other binary maximin problems exhibit a more challenging and sparse payoff landscape, causing the original methods inapplicable. To overcome this issue, we provide a variant of the level-based theorem, which takes the initialisation of algorithms into account[1]. Note that if m_2 is set to be 1, then Theorem 1 recovers the original theorem introduced by Lehre (2022).

Theorem 1. *Given subsets $A_j \subseteq \mathcal{X}, B_j \subseteq \mathcal{Y}$ for $j \in [m]$ where $A_1 = \mathcal{X}$ and $B_1 = \mathcal{Y}$, define $T := \min\{t\lambda \mid (P_t \times Q_t) \cap (A_m \times B_m) \neq \emptyset\}$, where for all $t \in \mathbb{N}, P_t \in \mathcal{X}^\lambda$ and $Q_t \in \mathcal{Y}^\lambda$ are the populations of Algorithm 1 in generation t. Denote the current level of the population by $j := \max\{\, i \in [m-1] \mid |(P \times Q) \cap (A_i \times B_i)| \geq \gamma_0 \lambda^2\}$ where $\gamma_0 \in (0, 1)$.*

Given $m_2 \in \mathbb{N}$ where $m_2 < m$, suppose $P_0 \times Q_0$ is initialised with the current level $j \geq m_2$. If there exist $z_{m_2}, \dots, z_{m-1}, \delta \in (0, 1)$ such that for any populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ with the current level $j \geq m_2$,

(G1) for $(x, y) \sim \mathcal{D}(P, Q)$, $\Pr(x \in A_{j+1}) \Pr(y \in B_{j+1}) \geq z_j$;
(G2a) for all $\gamma \in (0, \gamma_0)$, if $|(P \times Q) \cap (A_{j+1} \times B_{j+1})| \geq \gamma\lambda^2$, then for $(x, y) \sim \mathcal{D}(P, Q)$, $\Pr(x \in A_{j+1}) \Pr(y \in B_{j+1}) \geq (1 + \delta)\gamma$;
(G2b) for $(x, y) \sim \mathcal{D}(P, Q)$, $\Pr(x \in A_j) \Pr(y \in B_j) \geq (1 + \delta)\gamma_0$;
(G3) and the population size $\lambda \in \mathbb{N}$ satisfies $\lambda \geq c' \log(m/z_*)$ for a sufficiently large constant c', where $z_: = \min_{m_2 \leq i \leq m-1} z_i$.

then there exists a constant $c'' > 0$ such that any $r > 0$,

$$\Pr\left(T \geq r\frac{c''\lambda}{\delta}\left(\lambda^2(m - m_2) + \sum_{i=m_2}^{m-1} \frac{1}{z_i}\right)\right) \leq \frac{1 + o(1)}{r}.$$

Exponential Lower Bounds for RLS-PD, (1+1)-CoEA on Diagonal Game

In this section, we focus on the single-pair CoEA with pairwise dominance.

Algorithm 2: Single-Pair CoEA with Pairwise Dominance

Require: Payoff function $g : \{0, 1\}^n \times \{0, 1\}^n \to \mathbb{R}$.
Require: Mutation, `mutate` $: \{0, 1\}^{2n} \to \{0, 1\}^{2n}$.
Sample x, y uniformly at random from $\{0, 1\}^n$;
for $t \in \{1, 2, \dots\}$ **do**
  $(x', y') =$ `mutate`(x, y);
  if $(x', y') \succeq_g (x, y)$ **then** $(x, y) := (x', y')$;

[1]A similar modification can be found in (Hevia Fajardo and Lehre 2024).

In RLS-PD, the local mutation operator `mutate` creates x', y' by copying x and y, and flipping exactly one bit chosen uniformly at random from either x or y. In (1+1)-CoEA, the bit-wise mutation operator `mutate` creates x' and y' by copying x, y and flipping each bit independently with probability χ/n, where $\chi \in (0, n)$.

RLS-PD and DIAGONAL Game

The first competitive CoEA we consider is RLS-PD. We show that it fails to find the maximin optimum of DIAGONAL games in polynomial runtime.

RLS-PD samples both the design x and the test case y uniformly at random. Then, until the termination condition is met, either x or y is mutated by the local mutation operator. After that, RLS-PD selects the pair of designs and test cases based on pairwise dominance as defined in Definition 2. Unlike the success of RLS-PD on BILINEAR (Hevia Fajardo, Lehre, and Lin 2023), it fails to find the maximin optimum of DIAGONAL in polynomial runtime.

Theorem 2. *Consider RLS-PD on* DIAGONAL *and problem size* $n \in \mathbb{N}$. *The runtime of RLS-PD for finding the maximin optimum of* DIAGONAL *is at least* $e^{\Omega(n)}$ *with probability at least* $1 - e^{-\Omega(n)}$.

Theorem 2 shows that the algorithm performs a random walk before reaching the optimum. In this case, the selection mechanism fails to distinguish which individual is fitter if search points share the same payoff. Theorem 2 also suggests that DIAGONAL is a challenging and non-trivial maximin problem compared to BILINEAR since it takes polynomial runtime for RLS-PD to solve BILINEAR (Hevia Fajardo, Lehre, and Lin 2023) but at least exponential runtime to solve DIAGONAL.

There are various simple and effective mutation operators used in evolutionary algorithms (Doerr and Neumann 2019). RLS-PD uses the local mutation operator, which is one of the commonly used mutation operators in evolutionary algorithms. This operator picks any bit position uniformly at random and flips the selected bit. Another well-known mutation operator is the bit-wise mutation operator, which flips each bit in the bit-string with probability χ/n where the mutation rate $\chi \in (0, n)$ is some constant. The bit-wise mutation performs a more diverse search in the Hamming neighbourhood of the given search point. What if we modify the mutation operator from local mutation to bit-wise mutation? The bit-wise mutation operator potentially allows the search point to make more significant jumps in the search space at one iteration. Can the algorithm then find the optimum more efficiently? To answer these questions, we analyse the runtime of (1+1)-CoEA on DIAGONAL.

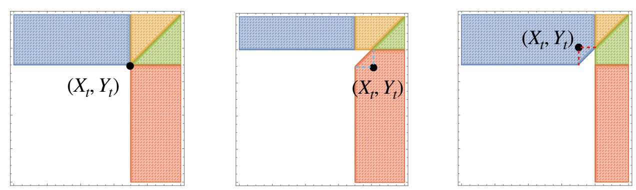
(1+1)-CoEA and DIAGONAL Game

We consider (1+1)-CoEA with the pairwise dominance relation. We want to explore the impact of changing the mutation operator compared to RLS-PD and whether this new variant can find the optimum more efficiently.

(1+1)-CoEA samples both the design x and the test case y uniformly at random. Then, until the termination criteria are met, unlike RLS-PD, both x and y are mutated by a bit-wise mutation operator. After that, (1+1)-CoEA selects the pair of designs and test cases based on the pairwise dominance relation from Definition 2. We use Lemma 1 to show condition (2) in Negative Drift Theorem (Oliveto and Witt 2012; Rowe and Sudholt 2012; Doerr and Neumann 2019).

Lemma 2. Consider (1+1)-CoEA with pairwise dominance relation on DIAGONAL and a constant mutation rate $\chi = O(1)$. Suppose we have a search point $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ in iteration $t \in \mathbb{N}$ and define $M_t := 2n - (X_t + Y_t)$. For any t, if $\varepsilon n < M_t \leq n/4$ for some constant $\varepsilon > 0$, then there exist some $r, \eta > 0$ s.t. for any $j \geq 0$, $\Pr(|M_t - M_{t+1}| \geq j \mid \mathcal{F}_t) \leq r/(1+\eta)^j$.

Lemma 2 shows that (1+1)-CoEA makes big jumps with exponential decay probability, which satisfies the step size condition in the Negative Drift Theorem. To see how to bound the negative drift from (1+1)-CoEA, we visualise Lemma 1. In Figure 1, the coloured regions represent the potential areas where the search point may move in the next iteration under the pairwise dominance relation. We consider the Manhattan distance between the current search point and the optimum, denoted as $M_t = 2n - (X_t + Y_t)$. Generally, negative drift occurs when the search point moves to the blue and red regions. Positive drift occurs when the search point moves to the orange and green regions.



(a) $X_t = Y_t$ (b) $X_t > Y_t$ (c) $X_t < Y_t$

Figure 1: Visualisation of Lemma 1. The colour area represents the search point $(x, y) \in \mathcal{X} \times \mathcal{Y}$ s.t. $(x, y) \succeq_g (x_t, y_t)$ where $X_t = |x_t|_1, Y_t = |y_t|_1$. The blue region corresponds to case (1), the orange region corresponds to case (2), the green region corresponds to case (3) and the red region corresponds to case (4) in Lemma 1.

Lemma 3. With the same setting as Theorem 3 and sufficiently large $n \geq 1$, for any t and $M_t := 2n - X_t - Y_t$, if $\varepsilon n < M_t \leq 2\varepsilon n$ for any constant $\varepsilon \in (0, 1/20]$, then there exists $\delta > 0$ such that $\mathrm{E}_t(M_t - M_{t+1}) \leq -\delta$.

Once we show the existence of negative drift in Lemma 3, we can use the Negative Drift Theorem with Lemma 2 to derive Theorem 3.

Theorem 3. *Consider (1+1)-CoEA with pairwise dominance relation on* DIAGONAL, *constant mutation parameter* $\chi \in (0, 1]$, *and problem size* $n \in \mathbb{N}$. *The runtime of (1+1)-CoEA for finding the maximin optimum of* DIAGONAL *is at least* $e^{\Omega(n)}$ *with probability at least* $1 - e^{-\Omega(n)}$.

Theorem 3 shows that (1+1)-CoEA also fails to find the desired optimum for DIAGONAL in polynomial time. The proof for Theorem 3 is more complex than that of Theorem 2 because the current search point of the (1+1)-CoEA can potentially make a significant jump in the search space, while RLS-PD only makes a 1-bit-step local search around the given search point.

From our analysis of RLS-PD and (1+1)-CoEA on DIAGONAL, Theorem 2 and Theorem 3 both show that with high probability (i.e., $1 - e^{-\Omega(n)}$), single-pair CoEAs with pairwise dominance cannot find the maximin optimum of DIAGONAL efficiently, regardless of which mutation operators the algorithms use. DIAGONAL consists only of binary values, resulting in a very flat payoff landscape in the search space, unlike problems with real-valued payoff functions, such as BILINEAR (Hevia Fajardo, Lehre, and Lin 2023; Hevia Fajardo and Lehre 2023). Such a payoff landscape is challenging for CoEAs.

PDCoEA Finds the NE of Diagonal Efficiently

Pairwise Dominance CoEA

From our previous analysis, we know that (1+1)-type CoEAs with pairwise dominance cannot solve DIAGONAL efficiently. We ask whether a competitive CoEA can solve this problem by changing its selection mechanism or increasing its population size. To address this question, we analyse the runtime of PDCoEA on DIAGONAL in the following section.

Algorithm 3: Pairwise Dominance CoEA (Lehre 2022)

Require: Payoff function $g : \{0,1\}^n \times \{0,1\}^n \to \mathbb{R}$.
Require: Population size λ and mutation rate $\chi \in (0, n]$.

```
1:  for i ∈ [λ]
2:      Sample P_0(i) uniformly at random from {0,1}^n;
3:      Sample Q_0(i) uniformly at random from {0,1}^n;
4:  for t ∈ {0,1,2,...} until termination criterion met do
5:      for i ∈ [λ] do
6:          Sample (x_1, y_1) ~ Unif(P_t × Q_t);
7:          Sample (x_2, y_2) ~ Unif(P_t × Q_t);
8:          if (x_1, y_1) ⪰_g (x_2, y_2) then (x, y) := (x_1, y_1)
9:          else (x, y) := (x_2, y_2)
10:         Obtain x' by flipping each bit in x with prob. χ/n;
11:         Obtain y' by flipping each bit in y with prob. χ/n;
12:         Set P_{t+1}(i) := x' and Q_{t+1}(i) := y'.
```

Algorithm 3 samples both the designs P_0 and the test cases Q_0 uniformly at random. Before mutation, Algorithm 3 uses 2-tournament selection to compare the pairs of designs and test cases. First, two pairs are sampled uniformly at random (lines 6-7), and then they are compared based on the pairwise dominance relation from Definition 2 (lines 8-9). Finally, both $x \in P_t$ and $y \in Q_t$ are mutated by a bit-wise mutation operator.

In this section, we use Theorem 1 to prove that PDCoEA can efficiently find the optimum of the DIAGONAL games. In general, level-based analysis partitions the whole search space into several levels and analyses the upgrade among different levels. Theorem 1 enables us to track the state of the algorithm and provides the tail bound on the number of function evaluations until the current population contains an individual hitting a target set (set to be our final level), given that conditions (G1)-(G3) hold. We defer the proofs of (G1)-(G3) conditions to the appendix.

Before showing our main result (Theorem 4), we define the following pixels and partitions that we will use in Theorem 1. We define the pixels: $j \in \{1, \cdots, n\}$,

$$A_j := \{x \in \{0,1\}^n \mid |x|_1 = j\} \text{ and } B_j := \{y \in \{0,1\}^n \mid |y|_1 = j\}. \quad (1)$$

Next, we define the following partitions that we will use in Theorem 1. Given $(U_0 \times V_0), \cdots, (U_n \times V_n)$, we define $U_0 = \mathcal{X}$ and $V_0 = \mathcal{Y}$ and for $j \in \{1, \cdots, n\}$, where we suppose $n = 4k$ (and thus $n/2 = 2k$ to simplify the notation during our calculation. One can use a different alternating partition for odd n)

$$U_{2j} := A_{2j} \text{ and } U_{2j+1} := A_{2j+1} \cup A_{2j+2}; \quad V_{2j} := B_{2j} \cup B_{2j+1} \text{ and } V_{2j+1} := B_{2j+1}. \quad (2)$$

Given two populations of solutions $P_t \in \mathcal{X}^\lambda, Q_t \in \mathcal{Y}^\lambda$, we define the current level by $j := \max\{i \in [n] \cup \{0\} \mid |(P_t \times Q_t) \cap (A_i \times B_i)| \geq \gamma_0 \lambda^2\}$ where $\gamma_0 \in (0,1)$ is constant and $\lambda \in \mathbb{N}$ is the population size. To estimate the runtime of a population-based evolutionary algorithm, we consider the number of function evaluations until the population contains at least one individual arriving at the final level (i.e. the first hitting time of the population of solution hitting the final level multiplied by the population size λ): $T := \inf\{\lambda t > 0 \mid (P_t \times Q_t) \cap (U_n \times V_n) \neq \emptyset\}$.

Technical Lemmas for Level-Based Analysis

In order to use Theorem 1, we need to satisfy (G1), (G2a), (G2b) and (G3). (G3) is easily satisfied by setting $\lambda = \Omega(\log n)$. So the rest of this section is to show conditions (G1), (G2a) and (G2b)[2].

Ensuring Condition (G1)

In this section, we compute the upgrade probability in condition (G1). It would be hard for $A \times B$ to upgrade to $A_1 \times B_1$ when the population falls off the diagonal. To resolve this, we assume the algorithm initialises the population at level $n/2$ and the population will keep hill-climbing along the diagonal from level $n/2$ to level n. It is also worth noting that the original tool (Level-Based Theorem (Lehre 2022)) does not consider the population's initialisation and thus cannot be directly applied to our problem.

Lemma 4. Given two populations P and Q in PDCoEA with population size $\lambda \in \text{poly}(n)$ and constant mutation rate $\chi \in (0, \ln 2/2)$ on DIAGONAL with problem size $n \in \mathbb{N}$, assume PDCoEA initialises P_0, Q_0 at $|P_0(i)|_1 = |Q_0(i)|_1 = n/2$ for all $i \in [\lambda]$. For the current level $j \geq n/2$, there exists $z_{n/2}, \cdots, z_{n-1} \in (0,1)$ such that for $(x,y) \sim \mathcal{D}(P,Q)$,

$$\Pr(x \in U_{j+1}) \cdot \Pr(y \in V_{j+1}) \geq z_j.$$

Lemma 4 shows that if the populations in PDCoEA are initialised at level $j = n/2$ and the current level is above $n/2$, then we can obtain a positive upgrade probability.

[2] We provide the visualisation of level partitions for PDCoEA on DIAGONAL in the appendix.

Ensuring Condition (G2a)

For PDCoEA, recall that $\mathcal{D}(P,Q)$ in Algorithm 1 is $\mathcal{D} = \text{SELECT} \circ \text{MUTATE}$, where SELECT corresponds to lines 6-9, and MUTATE corresponds to lines 10-12 in Algorithm 3. We denote the SELECT probability by $\Pr_{\text{select}}$ and the MUTATE probability by $\Pr_{\text{mutate}}$. Note that (G2a) in Theorem 1 keeps track of the multiplicative growth on the next current level. In the following analysis, without loss of generality, we assume the current level j is even to simplify the calculation. Before showing the next lemma, we need to make some assumptions: there exists a constant (with respect to χ) $\alpha \in (0,1)$ such that

$$\text{(A | G2a):} \Pr_{y \sim \text{Unif}(Q)}(y \in B_j) \geq \alpha.$$

This means that in each iteration, when the population is at the current level, there exists a constant fraction of the population in $A_j \times B_j$. Our current analysis works when assuming (A | G2a) holds. The other case where (A | G2a) does not hold is left as an open problem.

Lemma 5. Given two populations P and Q in PDCoEA with population size $\lambda \in \text{poly}(n)$ and a sufficiently small constant mutation rate $\chi \in (0,1)$ on DIAGONAL with problem size n, assume the assumption (A | G2a) holds with constant $\gamma_0 \geq \left(-1 + \sqrt{4\alpha(1+\delta)+1}\right)/2\alpha$ where $\delta \in (0,\alpha)$. For all $\gamma \in (0,\gamma_0)$, if $|(P \times Q) \cap (U_{j+1} \times V_{j+1})| \geq \gamma\lambda^2$, then for a sample $(x,y) \sim \mathcal{D}(P,Q)$, there exists a constant $\delta_0 > 0$ such that for all current levels $j \in [n/2, n]$, $\Pr(x \in U_{j+1})\Pr(y \in V_{j+1}) \geq (1+\delta_0)\gamma$.

Lemma 5 shows that we can obtain the multiplicative growth on the next level if the mutation rate is relatively low.

Ensuring Condition (G2b)

(G2b) in Theorem 1 keeps track of the multiplicative growth on the current level. Before showing (G2b) condition, we need to show that if the population starts at level $n/2$ (as stated in Theorem 4, with the assumption (A | G2b) holds, we can have the multiplicative growth in the current level. (A | G2b) means that there exists a certain fraction of the population in previous pixels for $x \in A_{<j}$ and $y \in B_{<j} \cup B_j$. To proceed with this, we show Lemma 6.

Lemma 6. Let $\mathcal{D}$ be the operator associated to Algorithms 3 (PDCoEA) on DIAGONAL. Let P and Q be populations of PDCoEA at current level $j \in [n/2, n]$ of DIAGONAL with respect to the partitions U, V and $\gamma_0 \in (0,1)$. There exists a constant $\delta_0 > 0$ and sufficiently small mutation rate χ, such that if for some constant $\delta > 0$, (A | G2b):

$$\Pr_{x \sim \text{Unif}(P)}(x \in A_{<j}) \left(\Pr_{y \sim \text{Unif}(Q)}(y \in B_{<j} \cup B_j)^2 + 1 \right) \geq \frac{1 - \gamma_0^2 + \delta}{\gamma_0} \text{ holds.}$$

then for $(x,y) \sim \mathcal{D}(P,Q)$, $\Pr(x \in U_j)\Pr(y \in V_j) \geq (1+\delta_0)\gamma_0$.

Next, we show that the condition (A | G2b) holds with high probability.

Lemma 7. Suppose PDCoEA initialises P_0, Q_0 at $|P_0(i)|_1 = |Q_0(i)|_1 = n/2$ for all $i \in [\lambda]$ with the runtime of Algorithm 3 finding the optimum $T \in \mathbb{N}$ and $(x_t, y_t) \sim \mathcal{D}(P_t, Q_t)$ at iteration $t \leq T$. With the same setting of Lemma 6, the assumption (A | G2b) with some constant $\delta \in (0, \chi e^{-\chi}/7)$: there exists $1 \geq \gamma_0 \geq \sqrt{(1+\delta)/(\chi e^{-\chi}/7 + 1)}$ such that (A | G2b) holds with probability $1 - e^{-\Omega(n)}$.

Lemma 8. Given two populations from PDCoEA, P and Q with population size $c \log n \leq \lambda \in \text{poly}(n)$ where $c > 0$ is some constant and low constant mutation rate χ on DIAGONAL with problem size n, assume constant $\gamma_0 \geq \sqrt{(1+\delta)/(\chi e^{-\chi}/7 + 1)}$ with constant $\delta \in (0, \chi e^{-\chi}/7)$ and sufficiently large n. Suppose PDCoEA initialises P_0, Q_0 at $|P_0(i)|_1 = |Q_0(i)|_1 = n/2$ for all $i \in [\lambda]$ with the runtime of Algorithm 3 finding the optimum $T \in \mathbb{N}$ and $(x_t, y_t) \sim \mathcal{D}(P_t, Q_t)$ at iteration $t \leq T$. There exists a constant $\delta > 0$ such that for all current levels $j \in [n/2, n]$, $\Pr(x_t \in U_j)\Pr(y_t \in V_j) \geq (1+\delta)\gamma_0$.

Finally, Lemma 8 states that if the mutation rate is set low, γ_0 large enough and furthermore the population is initialised at level $A_{n/2} \times B_{n/2}$, then we can obtain the multiplicative growth in our current level. Next, we conclude everything in the level-based argument using Theorem 1 with Lemma 4, 5 and 8.

Theorem 4. *Assume the condition (A | G2a) holds with* $\gamma_0 \geq \left(\sqrt{4\alpha(1+\delta)+1} - 1\right)/2\alpha$ *where* $\delta \in (0,\alpha)$ *and* $\alpha > 0$ *is some constant with respect to* χ*. Assume that for a sufficiently large constant c, it holds* $c \log n \leq \lambda \in \text{poly}(n)$ *and Algorithm 3 initialises* P_0, Q_0 *at* $|P_0(i)|_1 = |Q_0(i)|_1 = n/2$ *for all* $i \in [\lambda]$*. If* χ *is a sufficiently low constant, then there exists a constant (w.r.t* n *and* λ*)* $d > 0$ *such that the runtime of Algorithm 3 is at most* $rd\lambda n\left(\lambda^2 + n^2\right)$ *with probability at least* $1 - 1/r$*.*

Theorem 4 shows that if we set a low constant mutation rate and large population size, then PDCoEA can find the optimum of DIAGONAL in polynomial runtime with high probability. To the best of our knowledge, although various empirical studies demonstrate the existence of a "coevolutionary arms race" (Gomes, Mariano, and Christensen 2014; Ficici and Pollack 1998; Popovici et al. 2012; Hillis 1990), including recent applications of coevolutionary algorithms (Hemberg et al. 2021; Xue et al. 2023), this is the first provable coevolutionary "arms race" for solving binary maximin optimisation problems. This result not only demonstrates the promising potential of coevolution with pairwise dominance but also rigorously proves the necessity of large population sizes and low mutation rates to maintain a "coevolutionary arms race" on binary maximin optimisation problems.

In addressing the technical constraints of Theorem 4, it is important to note that generalising the result without the initialisation constraint is challenging. Proving Theorem 4 with random initialisation introduces a random walk phase, and it remains unclear whether existing tools can effectively address genetic drift toward the diagonal. The primary objective of this analysis is to rigorously demonstrate the “coevolutionary arms race,” and for this purpose, the current Theorem 4 sufficiently illustrates this point. We consider Theorem 4 as a foundational step for future advancements.

Experiments

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

Settings: We conduct the experiments with all CoEAs on the DIAGONAL problem with three different problem sizes: $n = 100$, 500, and 1000. Firstly, we conduct experiments of all CoEAs on DIAGONAL with $n = 100$, generating heatmaps for the empirical mean of runtimes under different configurations. Secondly, we explore the best possible configurations of χ and λ for PDCoEA with problem sizes $n = 100, 500$, and 1000, providing detailed statistics, including tables for the empirical mean of runtimes, p-values from Wilcoxon rank-sum tests for runtimes, and boxplots for runtime distribution with respect to low mutation rates $\chi \in [0.1, 0.4]$. Finally, we pick $\chi = 0.3$ and $\lambda = c \log n$ where $c = 10, 20, 30, 40, 50$, and 60 and conduct the experiments under problem sizes $n =$ 100-1000. The budget for each run is set to 10^8 function evaluations for $n = 100$ and 10^9 function evaluations for $n = 500$ and 1000. For each configuration, we conduct 100 independent runs.

Results: We defer other empirical results to the appendix and present part of our findings here. As shown in Figure 2, with respect to different problem sizes, once the population size reaches certain thresholds (i.e., $\lambda \geq 0.4n$ for $n = 100$ and $\lambda \geq 0.2n$ for $n = 500, 1000$) under a low mutation rate $\chi \in [0.1, 0.4]$. Although PDCoEA can solve DIAGONAL under certain configurations, a high mutation rate above $\chi = 0.5$ results in the inefficiency of PDCoEA. The data (in the appendix) shows that the best empirical mean of runtime, for $n = 100, 500, 1000$, is approximately 0.12×10^6, 1.05×10^6 and 2.71×10^6, respectively, under a certain low mutation rate $\chi = 0.3$. Moreover, Figure 3 suggests that although our theoretical bound is correct, it might not be tight enough in terms of asymptotic order. While our bounds are sufficient to show a distinct separation between CoEAs with a large population and those with a small population, our theoretical bounds can be improved.

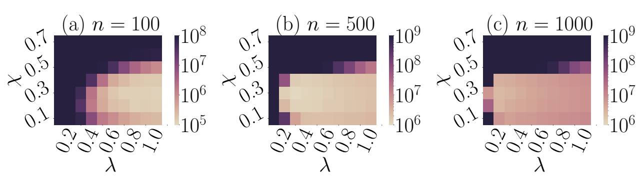


Figure 2: Heatmaps for the mean runtime of PDCoEA on Diagonal against different mutation rates and each pixel in the plot represented the empirical mean of runtime among 100 independent runs with respect to different mutation rates and population sizes. The mutation rate χ in the vertical axis ranges from 0.1 to 0.7, and the population size λ in the horizontal axis ranges from $0.1n$ to n.

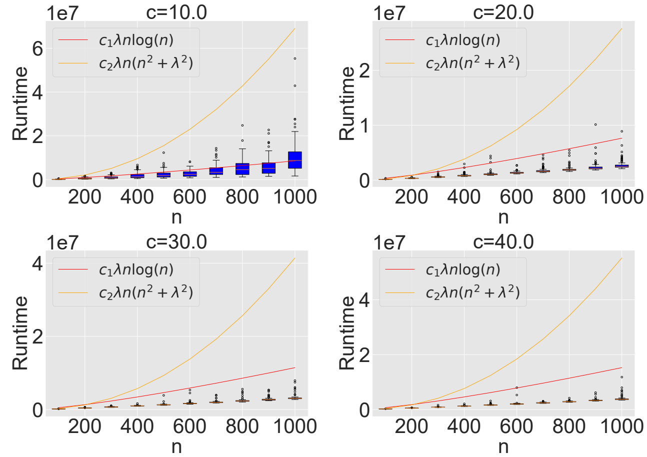


Figure 3: Boxplots for runtime of PDCoEA ($\chi = 0.3$) on DIAGONAL with respect to different c and problem size n.

Conclusion and Discussion

Discrete maximin optimisation is challenging. This paper presents the first runtime analysis of CoEAs with pairwise dominance on non-trivial b inary two-player zero-sum games. We show that for the DIAGONAL game, RLS-PD and (1+1)-CoEA require exponential time to find the optimum due to negative drift, highlighting the inefficiency of CoEAs without sufficiently large population sizes in flat payoff landscapes. However, PDCoEA with a low mutation rate and large population size $\lambda = \Omega(\log n)$ can efficiently find the optimum with high probability.

This paper presents a clear separation among various CoEAs on binary two-player zero-sum games, emphasising the need for large populations, as opposed to previous studies on BILINEAR (Hevia Fajardo, Lehre, and Lin 2023). It demonstrates the feasibility of runtime analysis for population-based competitive CoEAs, suggesting that these tools can broaden the scope of future analyses. Practically, it shows that single-pair CoEAs like RLS-PD and (1+1)-CoEA may be unsuitable for such problems, recommending instead CoEAs with large populations and low mutation rates for challenging, flat payoff landscapes.

This work marks a significant first step in the rigorous runtime analysis of population-based CoEAs on DIAGONAL and introduces a variant of the level-based theorem for coevolution. While our analysis is focused on a specific class of sparse binary games, we believe the level-based theorem has broader applicability to general coevolutionary processes. Extending this tool to more general partitions beyond the Cartesian product of search spaces $\mathcal{X}$ and $\mathcal{Y}$ would be valuable. Although our empirical results imply that the current bound may not be tight, we conjecture that the runtime of PDCoEA could be $O(\lambda n \log n)$ under certain conditions. Future research should explore a wider range of binary games, develop additional coevolutionary algorithms, and refine the runtime bounds for PDCoEA on DIAGONAL.

Acknowledgements

We would like to thank Dr Alistair Benford and Dr Mario Alejandro Hevia Fajardo for the fruitful discussion and comments on an earlier draft of this paper and also thank the anonymous reviewers for their helpful reviews. 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.

References

Auger, D.; Liu, J.; Ruette, S.; Saint-Pierre, D.; and Teytaud, O. 2015. Sparse Binary Zero-Sum Games. In *Asian Conference on Machine Learning*, 173–188. PMLR.

Benford, A.; and Lehre, P. K. 2024. Runtime Analysis of Coevolutionary Algorithms on a Class of Symmetric Zero-Sum Games. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '24. New York, NY, USA: Association for Computing Machinery.

Benford, A.; Olhofer, M.; Rodemann, T.; and Lehre, P. K. 2024. Bicriteria Optimisation of Average and Worst-Case Performance Using Coevolutionary Algorithms. In *IEEE Congress on Evolutionary Computation (IEEE CEC) 2024*. IEEE.

Beznosikov, A.; Sadiev, A.; and Gasnikov, A. 2020. Gradient-Free Methods with Inexact Oracle for Convex-Concave Stochastic Saddle-Point Problems. In *International Conference on Mathematical Optimization Theory and Operations Research*, 105–119. Springer.

Colson, B.; Marcotte, P.; and Savard, G. 2007. An Overview of Bilevel Optimization. *Annals of Operations Research*, 153: 235–256.

Daskalakis, C.; and Panageas, I. 2018. The Limit Points of (Optimistic) Gradient Descent in Min-Max Optimization. *Advances in Neural Information Processing Systems*, 31.

Do, A. V.; Neumann, A.; Neumann, F.; and Sutton, A. M. 2023. Rigorous Runtime Analysis of MOEA/D for Solving Multi-Objective Minimum Weight Base Problems. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Doerr, B.; and Neumann, F. 2019. *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*.

Doerr, B.; and Qu, Z. 2023. Runtime Analysis for the NSGA-II: Provable Speed-Ups from Crossover. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 12399–12407. AAAI Press.

Dvinskikh, D.; Tominin, V.; Tominin, Y.; and Gasnikov, A. 2022. Gradient-Free Optimization for Non-Smooth Saddle Point Problems under Adversarial Noise. *arXiv preprint arXiv:2202.06114*.

Eiben, A.; and Smith, J. 2015. *Introduction to Evolutionary Computing*. Natural Computing Series. Berlin, Heidelberg: Springer Berlin Heidelberg.

Ficici, S. G.; and Pollack, J. B. 1998. Challenges in Coevolutionary Learning, Arms-Race Dynamics: Open-Endedness, and Mediocre Stable States. In *Proceedings of the Sixth International Conference on Artificial Life*, 238–247. MIT Press Cambridge, MA.

Gasnikov, A.; Novitskii, A.; Novitskii, V.; Abdukhakimov, F.; Kamzolov, D.; Beznosikov, A.; Takáč, M.; Dvurechensky, P.; and Gu, B. 2022. The Power of First-Order Smooth Optimization for Black-Box Non-Smooth Problems. *arXiv preprint arXiv:2201.12289*.

Gidel, G.; Jebara, T.; and Lacoste-Julien, S. 2017. Frank-Wolfe Algorithms for Saddle Point Problems. In *Artificial Intelligence and Statistics*, 362–371. PMLR.

Gomes, J.; Mariano, P.; and Christensen, A. L. 2014. Novelty Search in Competitive Coevolution. In *Parallel Problem Solving from Nature–PPSN XIII: 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings 13*, 233–242. Springer.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 27.

Grand-Clément, J.; and Kroer, C. 2021. Conic Blackwell Algorithm: Parameter-Free Convex-Concave Saddle-Point Solving. *Advances in Neural Information Processing Systems*, 34: 9587–9599.

He, J.; and Yao, X. 2001. Drift Analysis and Average Time Complexity of Evolutionary Algorithms. *Artificial Intelligence*, 127(1): 57–85.

Hemberg, E.; Toutouh, J.; Al-Dujaili, A.; Schmiedlechner, T.; and O'Reilly, U.-M. 2021. Spatial Coevolution for Generative Adversarial Network Training. *ACM Transactions on Evolutionary Learning and Optimization*, 1(2): 1–28.

Hevia Fajardo, M.; Lehre, P. K.; Toutouh, J.; Hemberg, E.; and O'Reilly, U.-M. 2024. Analysis of a Pairwise Dominance Coevolutionary Algorithm with Spatial Topology. In *Genetic Programming Theory and Practice XX*, 19–44. Springer.

Hevia Fajardo, M.; and Lehre, P. K. 2024. Ranking Diversity Benefits Coevolutionary Algorithms on an Intransitive Game. In *International Conference on Parallel Problem Solving from Nature*, 213–229. Springer.

Hevia Fajardo, M. A.; and Lehre, P. K. 2023. How Fitness Aggregation Methods Affect the Performance of Competitive CoEAs on Bilinear Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23. New York, NY, USA: Association for Computing Machinery.

Hevia Fajardo, M. A.; Lehre, P. K.; and Lin, S. 2023. Runtime Analysis of a Co-Evolutionary Algorithm: Overcoming Negative Drift in Maximin-Optimisation. In *Proceedings of the 17th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, FOGA '23, 73–83. New York, NY, USA: Association for Computing Machinery.

Hillis, W. 1990. Co-evolving Parasites Improve Simulated Evolution as an Optimization Procedure. *Physica D: Nonlinear Phenomena*, 42(1-3): 228–234.

Lehre, P. K. 2010. Negative Drift in Populations. In *International Conference on Parallel Problem Solving from Nature*, 244–253. Springer.

Lehre, P. K. 2022. 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, 1408–1416.

Lehre, P. K.; and Lin, S. 2024. Overcoming Binary Adversarial Optimisation with Competitive Coevolution. In *Proceedings of the 18th International Conference on Parallel Problem Solving From Nature PPSN*, 18.

Lin, T.; Jin, C.; and Jordan, M. 2020. On Gradient Descent Ascent for Nonconvex-Concave Minimax Problems. In *International Conference on Machine Learning*, 6083–6093. PMLR.

Mertikopoulos, P.; Lecouat, B.; Zenati, H.; Foo, C.-S.; Chandrasekhar, V.; and Piliouras, G. 2018. Optimistic Mirror Descent in Saddle-Point Problems: Going the Extra (Gradient) Mile. *arXiv preprint arXiv:1807.02629*.

Oliveto, P. S.; and Witt, C. 2012. Erratum: Simplified Drift Analysis for Proving Lower Bounds in Evolutionary Computation. ArXiv:1211.7184 [cs].

O'Hanley, J. R.; and Church, R. L. 2011. Designing Robust Coverage Networks to Hedge Against Worst-Case Facility Losses. *European Journal of Operational Research*, 209(1): 23–36.

Palaniappan, B.; and Bach, F. 2016. Stochastic Variance Reduction Methods for Saddle-Point Problems. *Advances in Neural Information Processing Systems*, 29.

Pollack, J. B.; and Blair, A. D. 1998. Co-evolution in the Successful Learning of Backgammon Strategy. *Machine Learning*, 32: 225–240.

Popovici, E.; Bucci, A.; Wiegand, R. P.; and De Jong, E. D. 2012. Coevolutionary Principles. In Rozenberg, G.; Bäck, T.; and Kok, J. N., eds., *Handbook of Natural Computing*, 987–1033. Berlin, Heidelberg: Springer Berlin Heidelberg.

Robey, A.; Latorre, F.; Pappas, G. J.; Hassani, H.; and Cevher, V. 2024. Adversarial Training Should Be Cast as a Non-Zero-Sum Game. In *The Twelfth International Conference on Learning Representations*.

Rowe, J. E.; and Sudholt, D. 2012. The Choice of the Offspring Population Size in the (1,λ) EA. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, 1349–1356. New York, NY, USA: Association for Computing Machinery.

von Neumann, J. 1928. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1): 295.

Wang, R.; Lehman, J.; Clune, J.; and Stanley, K. O. 2019. Poet: Open-Ended Coevolution of Environments and Their Optimized Solutions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 142–151.

Wang, R.; Lehman, J.; Rawal, A.; Zhi, J.; Li, Y.; Clune, J.; and Stanley, K. 2020. Enhanced POET: Open-Ended Reinforcement Learning Through Unbounded Invention of Learning Challenges and Their Solutions. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 9940–9951. PMLR.

Wei, C.-Y.; Lee, C.-W.; Zhang, M.; and Luo, H. 2021. Linear Last-Iterate Convergence in Constrained Saddle-Point Optimization. In *International Conference on Learning Representations*.

Xue, K.; Qian, C.; Xu, L.; and Fei, X. 2021. Evolutionary Gradient Descent for Non-Convex Optimization. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*.

Xue, K.; Wang, R.-J.; Li, P.; Li, D.; Jianye, H.; and Qian, C. 2023. Sample-Efficient Quality-Diversity by Cooperative Coevolution. In *The Twelfth International Conference on Learning Representations*.

Zhang, B. H.; and Sandholm, T. 2024. Exponential Lower Bounds on the Double Oracle Algorithm in Zero-Sum Games. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*. International Joint Conferences on Artificial Intelligence Organization.

Zhang, Y.; Zhang, G.; Khanduri, P.; Hong, M.; Chang, S.; and Liu, S. 2022. Revisiting and Advancing Fast Adversarial Training Through the Lens of Bi-Level Optimization. In *International Conference on Machine Learning*, 26693–26712. PMLR.

Zheng, W.; and Doerr, B. 2023. From Understanding Genetic Drift to a Smart-Restart Mechanism for Estimation-of-Distribution Algorithms. *Journal of Machine Learning Research*, 24(292): 1–40.

Zheng, W.; Liu, Y.; and Doerr, B. 2022. A First Mathematical Runtime Analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10408–10416.