Runtime analysis of competitive co-evolutionary algorithms for maximin optimisation of a bilinear function

Lehre, Per Kristian

DOI:
10.1145/3512290.3528853

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Lehre, PK 2022, Runtime analysis of competitive co-evolutionary algorithms for maximin optimisation of a bilinear function. in JE Fieldsend (ed.), *GECCO '22: Proceedings of the Genetic and Evolutionary Computation Conference.* GECCO: Genetic and Evolutionary Computation Conference, Association for Computing Machinery (ACM), New York, pp. 1408–1416, GECCO '22: Genetic and Evolutionary Computation Conference, Boston, Massachusetts, United States, 9/07/22. https://doi.org/10.1145/3512290.3528853

Link to publication on Research at Birmingham portal

Runtime Analysis of Competitive co-Evolutionary Algorithms for Maximin Optimisation of a Bilinear Function[*]

Per Kristian Lehre

School of Computer Science
University of Birmingham
United Kingdom

April 19, 2022

Abstract

Co-evolutionary algorithms have a wide range of applications, such as in hardware design, evolution of strategies for board games, and patching software bugs. However, these algorithms are poorly understood and applications are often limited by pathological behaviour, such as loss of gradient, relative over-generalisation, and mediocre objective stasis. It is an open challenge to develop a theory that can predict when co-evolutionary algorithms find solutions efficiently and reliably.

This paper provides a first step in developing runtime analysis for population-based competitive co-evolutionary algorithms. We provide a mathematical framework for describing and reasoning about the performance of co-evolutionary processes. An example application of the framework shows a scenario where a simple co-evolutionary algorithm obtains a solution in polynomial expected time. Finally, we describe settings where the co-evolutionary algorithm needs exponential time with overwhelmingly high probability to obtain a solution.

1 Introduction

Many real-world optimisation problems feature a strategic aspect, where the solution quality depends on the actions of other – potentially adversarial – players. There is a need for adversarial optimisation algorithms that operate under realistic assumptions. Departing from a traditional game theoretic setting, we assume two classes of players, choosing strategies from "strategy spaces" $\mathcal{X}$ and

[*] To appear in Proceedings of Genetic and Evolutionary Computation Conference (GECCO '22).

$\mathcal{Y}$ respectively. The objectives of the players are to maximise their individual "payoffs" as given by payoff functions $f, g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.

A fundamental algorithmic assumption is that there is insufficient computational resources available to exhaustively explore the strategy spaces $\mathcal{X}$ and $\mathcal{Y}$. In a typical real world scenario, a strategy could consist of making n binary decisions. This leads to exponentially large and discrete strategy spaces $\mathcal{X} = \mathcal{Y} = \{0, 1\}^n$. Furthermore, we can assume that the players do not have access to or the capability to understand the payoff functions. However, it is reasonable to assume that players can make repeated queries to the payoff function Fearnley and Savani (2016). Together, these assumptions render many existing approaches impractical, e.g., Lemke-Howson, best response dynamics, mathematical programming, or gradient descent-ascent.

Co-evolutionary algorithms (CoEAs) (see (Popovici et al., 2012) for a survey) could have a potential in adversarial optimisation, partly because they make less strict assumptions than the classical methods. Two populations are co-evolved (say one in $\mathcal{X}$, the other in $\mathcal{Y}$), where individuals are selected for reproduction if they interact successfully with individuals in the opposite population (e.g. as determined by the payoff functions f, g). The hoped for outcome is that an artificial "arms race" emerges between the populations, leading to increasingly sophisticated solutions. In fact, the literature describe several successful applications, including design of sorting networks Hillis (1990), software patching Arcuri and Yao (2008), and problems arising in cyber security O'Reilly et al. (2020).

It is common to separate co-evolution into co-operative and competitive co-evolution. Co-operative co-evolution is attractive when the problem domain allows a natural division into sub-components. For example, the design of a robot can be separated into its morphology and its control Pollack et al. (2001). A cooperative co-evolutionary algorithm works by evolving separate "species", where each species is responsible for optimising one sub-component of the overall solution. To evaluate the fitness of a sub-component, it is combined with sub-components from the other species to form a complete solution. Ideally, there will be a selective pressure for the species to cooperate, so that they together produce good overall designs Potter and Jong (2000).

The behaviour of CoEAs can be abstruse, where pathological population behaviour such as loss of gradient, focusing on the wrong things, and relativism Watson and Pollack (2001) prevent effective applications. It has been a long-standing open problem to develop a theory that can explain and predict the performance of co-evolutionary algorithms (see e.g. Section 4.2.2 in Popovici et al. (2012)), notably runtime analysis. Runtime analysis of EAs Doerr and Neumann (2020) have provided mathematically rigorous statements about the runtime distribution of evolutionary algorithms, notably how the distribution depends on characteristics of the fitness landscape and the parameter settings of the algorithm.

The only rigorous runtime analysis of co-evolution the author is aware of focuses on co-operative co-evolution. In a pioneer study, Jansen and Wiegand considered the common assumption that co-operative co-evolution allows a speedup for *separable* problems Jansen and Wiegand (2004). They compared rigorously the runtime of the co-operative co-evolutionary (1+1) Evolutionary Algorithm (CC (1+1) EA) with the classical (1+1) EA. Both algorithms follow the same template: They keep the single best solution seen so far, and iteratively produce new candidate solution by "mutating" the best solution. However, the algorithms use different mutation operators. The CC (1+1) EA restricts mutation to the bit-positions within one out of k blocks in each iteration. The choice of the current block alternates deterministically in each iteration, such that in k iterations, every block has been active once. The main conclusion from their analysis is that problem separability is not a sufficient criterion to determine whether the CC (1+1) EA performs better than the (1+1) EA. In particular, there are separable problems where the (1+1) EA outperforms the CC (1+1) EA, and there are inseparable problems where the converse holds. What the authors find is that CC (1+1) EA is advantageous when the problem separability matches the partitioning in the algorithm, and there is a benefit from increased mutation rates allowed by the CC (1+1) EA.

Much work remains to develop runtime analysis of co-evolution. Co-operative co-evolution can be seen as a particular approach to traditional optimisation, where the goal is to maximise a given objective function. In contrast, competitive co-evolutionary algorithms are employed for a wide range of solution concepts Ficici (2004). It is unclear to what degree results about co-operative CoEAs can provide insights about competitive CoEAs. Finally, the existing runtime analysis considers the CC (1+1) EA which does not have a population. However, it is particularly important to study co-evolutionary population dynamics to understand the pathologies of existing CoEAs.

This paper makes the following contributions: Section 2 introduces a generic mathematical framework to describe a large class of co-evolutionary processes. We then discuss how the population-dynamics of these processes can be described by a stochastic process. Section 3 defines "runtime" in the context of generic co-evolutionary processes and presents an analytical tool (a co-evolutionary level-based theorem) which can be used to derive upper bounds on the expected runtime. Section 4 specialises the problem setting to maximin-optimisation, and introduces a theoretical benchmark problem Bilinear which we envisage could play the role of OneMax for traditional runtime analysis. Section 5 introduces the algorithm PD-CoEA which is a particular co-evolutionary process tailored to maximin-optimisation. We then analyse the runtime of PD-CoEA on Bilinear using the level-based theorem, showing that there are settings where the algorithm obtains a solution in polynomial time. Finally, in Section 6 we demonstrate that the PD-CoEA possesses an "error threshold", i.e., a mutation rate above which the runtime is exponential for any problem. Due to space limitations, substantial parts of the technical analysis have been moved to the appendix.

1.1 Preliminaries

For any natural number $n \in \mathbb{N}$, we define $[n] := \{1, 2, \ldots, n\}$ and $[0..n] := \{0\} \cup [n]$. For a filtration $(\mathscr{F}_t)_{t \in \mathbb{N}}$ and a random variable X we use the shorthand notation $\mathbb{E}_t[X] := \mathbb{E}[X \mid \mathscr{F}_t]$. A random variable X is said to *stochastically dominate* a random variable Y, denoted $X \succeq Y$, if and only if $\Pr(Y \leq z) \geq \Pr(X \leq z)$ for all $z \in \mathbb{R}$. The Hamming distance between two bitstrings x and y is denoted $H(x, y)$. For any bitstring $z \in \{0,1\}^n$, $|z| := \sum_{i=1}^{n} z_i$, denotes the number of 1-bits in z.

2 Co-Evolutionary Algorithms

This section describes in mathematical terms a broad class of co-evolutionary processes (Algorithm 1). The definition takes inspiration from level-processes (see Algorithm 1 in Corus et al. (2018)) used to describe non-elitist evolutionary algorithms.

Algorithm 1 Co-evolutionary Process

```
Require: Population size $\lambda \in \mathbb{N}$ and 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 $Q_{t+1}(i) := y$.
    end for
end for
```

We assume that in each generation, the algorithm has two[1] populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ which we sometimes will refer to as the "predators" and the "prey". We posit that in each generation, the populations interact λ times, where each interaction produces in a stochastic fashion one new predator $x \in \mathcal{X}$ and one new prey $y \in \mathcal{Y}$. The interaction is modelled as a probability distribution $\mathcal{D}(P, Q)$ over $\mathcal{X} \times \mathcal{Y}$ that depends on the current populations. For a given instance of the framework, the operator $\mathcal{D}$ encapsulates all aspects that take place in producing new offspring, such as pairing of individuals, selection, mutation, crossover, etc. (See Section 5 for a particular instance of $\mathcal{D}$).

As is customary in the theory of evolutionary computation, the definition of the algorithm does not state any termination criterion. The justification for this omission is that the choice of termination criterion does not impact the definition of runtime we will use.

Notice that the predator and the prey produced through one interaction are not necessarily independent random variables. However, each of the λ interactions in one generation are independent and identically distributed random variables.

[1] The framework can be generalised to more populations.

2.1 Tracking the algorithm state

We will now discuss how the state of Algorithm 1 can be captured with a stochastic process. To determine the trajectory of a co-evolutionary algorithm, it is insufficient to track only one of the populations, as the dynamics of the algorithm is determined by the relationship between the two populations.

We shall see that it will be convenient to describe the state of the algorithm via the Cartesian product $P_t \times Q_t$. In particular, for subsets $A \subset \mathcal{X}$ and $B \subset Y$, we will study the drift of the stochastic process $Z_t := |(P_t \times Q_t) \cap (A \times B)|$. Naturally, there will be multiple probability dependencies among the λ^2 pairs in the product $P_t \times Q_t$. In order to not have to explicitly take these dependencies into account later in the paper, we now characterise properties of the distribution of Z_t in Lemma 1.

Lemma 1. *Given subsets $A \subset \mathcal{X}, B \subset \mathcal{Y}$, assume that for any $\delta > 0$ and $\gamma \in (0,1)$, the sample $(x,y) \sim \mathcal{D}(P_t, Q_t)$ satisfies*

$$\Pr(x \in A) \Pr(y \in B) \geq (1+\delta)\gamma.$$

Then the random variable $Z_{t+1} := |(P_{t+1} \times Q_{t+1}) \cap A \times B|$ satisfies

1) $\mathbb{E}_t[Z_{t+1}] \geq \lambda(\lambda-1)(1+\delta)\gamma$.

2) $\mathbb{E}_t\left[e^{-\eta Z_{t+1}}\right] \leq e^{-\eta\lambda(\gamma\lambda-1)}$ *for* $0 < \eta \leq (1-(1+\delta)^{-1/2})/\lambda$

3) $\Pr_t(Z_{t+1} < \lambda(\gamma\lambda-1)) \leq e^{-\delta_1\gamma\lambda\left(1-\sqrt{\frac{1+\delta_1}{1+\delta}}\right)}$ *for* $\delta_1 \in (0,\delta)$.

Proof. In generation $t+1$, the algorithm samples independently and identically λ pairs $(P_{t+1}(i), Q_{t+1}(i))_{i\in[\lambda]}$ from distribution $\mathcal{D}(P_t, Q_t)$. For all $i \in [\lambda]$, define the random variables $X_i' := \mathbb{1}_{\{P_{t+1}(i)\in A\}}$ and $Y_i' := \mathbb{1}_{\{Q_{t+1}(i)\in B\}}$. Then since the algorithm samples each pair $(P_{t+1}(i), Q_{t+1}(i))$ independently, and by the assumption of the lemma, there exists $p, q \in (0,1]$ such that $X' := \sum_{i=1}^{\lambda} X_i' \sim \text{Bin}(\lambda, p)$, and $Y' := \sum_{i=1}^{\lambda} Y_i' \sim \text{Bin}(\lambda, q)$, where $pq \geq \gamma(1+\delta)$. By these definitions, it follows that $Z_{t+1} = X'Y'$.

Note that X' and Y' are not necessarily independent random variables because X_i' and Y_i' are not necessarily independent. However, by defining two independent binomial random variables $X \sim \text{Bin}(\lambda, p)$, and $Y \sim \text{Bin}(\lambda, q)$, we readily have the stochastic dominance relation

$$Z_{t+1} = X'Y' \succeq XY - \sum_{i=1}^{\lambda} X_i Y_i. \tag{1}$$

The first statement of the lemma is now obtained by exploiting (1), Lemma 27, and the independence between X and Y

$$\mathbb{E}_t[Z_{t+1}] \geq \mathbb{E}\left[XY - \sum_{i=1}^{\lambda} X_i Y_i\right] = \mathbb{E}[X]\mathbb{E}[Y] - \sum_{i=1}^{\lambda} \mathbb{E}[X_i]\mathbb{E}_t[Y_i]$$
$$= p\lambda q\lambda - \lambda pq = pq\lambda(\lambda-1) \geq (1+\delta)\gamma\lambda(\lambda-1).$$

For the second statement, we apply Lemma 18 wrt X, Y, and the parameters $\sigma := \sqrt{1+\delta} - 1$ and $z := \gamma$. By the assumption on p and q, we have $pq \geq (1+\delta)\gamma = (1+\sigma)^2 z$, furthermore the constraint on parameter η gives

$$\eta \leq \frac{1}{\lambda}\left(1 - \frac{1}{\sqrt{1+\delta}}\right) = \frac{\sqrt{1+\delta}-1}{\lambda\sqrt{1+\delta}} = \frac{\sigma}{(1+\sigma)\lambda}.$$

The assumptions of Lemma 18 are satisfied, and we obtain from (1)

$$\begin{aligned}\mathbb{E}_t\left[e^{-\eta Z_{t+1}}\right] &\leq \mathbb{E}\left[\exp\left(-\eta XY + \eta\sum_{i=1}^{\lambda} X_i Y_i\right)\right] \\ &< e^{\eta\lambda} \cdot \mathbb{E}\left[e^{-\eta XY}\right] < e^{\eta\lambda} \cdot e^{-\eta\gamma\lambda^2} = e^{-\eta\lambda(\gamma\lambda-1)}.\end{aligned}$$

Given the second statement, the third statement will be proved by a standard Chernoff-type argument. Define $\delta_2 > 0$ such that $(1+\delta_1)(1+\delta_2) = 1+\delta$. For

$$\eta := \frac{1}{\lambda}\left(1 - \frac{1}{\sqrt{1+\delta_2}}\right) = \frac{1}{\lambda}\left(1 - \sqrt{\frac{1+\delta_1}{1+\delta}}\right)$$

and $a := \lambda(\gamma\lambda - 1)$, it follows by Markov's inequality

$$\begin{aligned}\Pr{}_t\left(Z_{t+1} \leq a\right) &= \Pr{}_t\left(e^{-\eta Z_{t+1}} \geq e^{-\eta a}\right) \leq e^{\eta a} \cdot \mathbb{E}_t\left[e^{-\eta Z_{t+1}}\right] \\ &\leq e^{\eta a} \cdot \exp\left(-\eta\lambda(\gamma(1+\delta_1)\lambda - 1)\right) \\ &= e^{\eta a - \eta a - \eta\gamma\lambda^2\delta_1} = e^{-\eta\gamma\lambda^2\delta_1} \\ &= \exp\left(-\delta_1\left(1 - \sqrt{\frac{1+\delta_1}{1+\delta}}\right)\gamma\lambda\right),\end{aligned}$$

where the last inequality applies statement 2. □

The next lemma is a variant of Lemma 1, and will be used to compute the probability of producing individuals in "new" parts of the product space $\mathcal{X} \times \mathcal{Y}$ (see condition (G1) of Theorem 3).

Lemma 2. *For $A \subset \mathcal{X}$ and $B \subset \mathcal{Y}$ define*

$$r := \Pr\left((P_{t+1} \times Q_{t+1}) \cap A \times B \neq \emptyset\right).$$

If for $(x, y) \sim \mathcal{D}(P_t, Q_t)$, it holds $\Pr(x \in A)\Pr(y \in B) \geq z$, then

$$\frac{1}{r} < \frac{3}{z(\lambda-1)} + 1.$$

Proof. Define $p := \Pr(x \in A), q := \Pr(y \in B)$ and $\lambda' := \lambda - 1$. Then by the definition of r and Lemma 25

$$\begin{aligned} r &\geq \Pr\left(\exists k \neq \ell \text{ s.t. } P_{t+1}(k) = u \wedge Q_{t+1}(\ell) = v\right) \\ &\geq (1-(1-p)^{\lambda})(1-(1-q)^{\lambda'}) > \left(\frac{\lambda' p}{1+\lambda' p}\right)\left(\frac{\lambda' q}{1+\lambda' q}\right) \\ &\geq \frac{\lambda'^2 z}{1+\lambda'(p+q)+\lambda' z} \geq \frac{\lambda'^2 z}{1+2\lambda'+\lambda'^2 z}.\end{aligned}$$

Finally, $\frac{1}{r} \leq \frac{2}{z\lambda'} + \frac{1}{z\lambda'^2} + 1 < \frac{3}{z\lambda'} + 1$. □

3 A Level-based Theorem for Co-Evolutionary Processes

This section defines a notion of runtime for Algorithm 1, and provides a generic tool (Theorem 3) for deriving upper bounds on the runtime. The proof of this theorem has been moved to Section A.

We will restrict ourselves to solution concepts that can be characterised as finding a given target subset $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y}$. This captures for example maximin optimisation or finding pure Nash equilibria. Within this context, the goal of Algorithm 1 is now to obtain populations P_t and Q_t such that their product intersects with the target set $\mathcal{S}$. We then define the runtime of an algorithm A as the number of interactions before the target subset has been found.

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

We follow the convention in analysis of population-based EAs that the granularity of the runtime is in generations, i.e., multiples of λ. The definition overestimates the number of interactions before a solution is found by at most $\lambda - 1$.

We now present a level-based theorem for co-evolution, which is one of the main contributions of this paper. The theorem states four conditions (G1), (G2a), (G2b), and (G3) which when satisfied imply an upper bound on the runtime of the algorithm. To apply the theorem, it is necessary to provide a sequence $(A_j \times B_j)_{j\in[m]}$ of subsets of $\mathcal{X} \times \mathcal{Y}$ called levels, where $A_1 \times B_1 = \mathcal{X} \times \mathcal{Y}$, and where $A_m \times B_m$ is the target set. It is recommended that this sequence overlaps to some degree with the trajectory of the algorithm. The "current level" j corresponds to the latest level occupied by at least a γ_0-fraction of the pairs in $P_t \times Q_t$. Condition (G1) states that the probability of producing a pair in the next level is strictly positive. Condition (G2a) states that the proportion of pairs in the next level should increase by a multiplicative factor larger than 1. Condition (G2a) implies that the fraction of pairs in the current level should not decrease below γ_0. Finally, Condition (G3) states a requirement in terms of the population size.

In order to make the "current level" of the populations well defined, we need to ensure that for all populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$, there exists at least one level $j \in [m]$ such that $|(P \times Q) \cap (A_j \times B_j)| \geq \gamma_0 \lambda^2$. This is ensured by defining an initial level $A_1 \times B_1 := \mathcal{X} \times \mathcal{Y}$.

Notice that the notion of "level" here is more general than in the classical level-based theorem Corus et al. (2018), in that they do not need to form a partition of the search space.

Finally, in this initial work, we have not made an effort in optimising the expression for the expected runtime. We conjecture that the dependency on λ can be reduced below λ^3, and that the leading constant c'' is relatively small.

Theorem 3. *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. If there exist $z_1, \ldots, z_{m-1}, \delta \in (0,1]$, and $\gamma_0 \in (0,1)$ such that for any populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ with "current level" $j := \max\{i \in [m-1] \mid |(P \times Q) \cap (A_i \times B_i)| \geq \gamma_0\lambda^2\}$*

(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 for a sufficiently large constant c', where $z_* := \min_{i \in [m-1]} z_i$,*

$$\lambda \geq c' \log(m/z_*),$$

then for a constant $c'' > 0$, $\mathbb{E}[T] \leq c''\lambda \left(\lambda^2 m + \sum_{i=1}^{m-1} 1/z_i\right)$.

4 Maximin Optimisation of Bilinear Functions

4.1 Maximin Optimisation Problems

This section introduces maximin-optimisation problems which is an important domain for competitive co-evolutionary algorithms Jensen (2004); Al-Dujaili et al. (2019); Miyagi et al. (2021). We will then describe a class of maximin-optimisation problems called Bilinear.

It is a common scenario in real-world optimisation that the quality of candidate solutions depend on the actions taken by some adversary. Formally, we can assume that there exists a function

$$g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R},$$

where $g(x, y)$ represents the "quality" of solution x when the adversary takes action y.

A cautious approach to such a scenario is to search for the candidate solution which maximises the objective, assuming that the adversary takes the least favourable action for that solution. Formally, this corresponds to the *maximin optimisation problem*, i.e., to maximise the function

$$f(x) := \min_{y \in \mathcal{Y}} g(x, y). \tag{2}$$

It is desirable to design good algorithms for such problems because they have important applications in economics, computer science, machine learning (GANs), and other disciplines.

However, maximin-optimisation problems are computationally challenging because to accurately evaluate the function $f(x)$, it is necessary to solve a minimisation problem. Rather than evaluating f directly, the common approach is to simultaneously maximise $g(x, y)$ with respect to x, while minimising $g(x, y)$ with respect to y. For example, if the gradient of g is available, it is popular to do gradient ascent-gradient descent.

Following conventions in theory of evolutionary computation Droste et al. (2006), we will assume that an algorithm has *oracle access* to the function g. This means that the algorithm can evaluate the function $g(x, y)$ for any selected pair of arguments $(x, y) \in \mathcal{X} \times \mathcal{Y}$, however it does not have access to any other information about g, including its definition or the derivative. Furthermore, we will assume that $\mathcal{X} = \mathcal{Y} = \{0, 1\}^n$. To develop a co-evolutionary algorithm for maximin-optimisation, we will rely on the following dominance relation on the set of pairs $\mathcal{X} \times \mathcal{Y}$.

Definition 2. *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) wrt g, denoted $(x_1, y_1) \succeq_g (x_2, y_2)$, if and only if*

$$g(x_1, y_2) \geq g(x_1, y_1) \geq g(x_2, y_1).$$

4.2 The Bilinear Problem

In order to develop appropriate analytical tools to analyse the runtime of evolutionary algorithms, it is necessary to start the analysis with simple and well-understood problems Wegener (2002). We therefore define a simple class of a maximin-optimisation problems that has a particular clear structure. The maximin function is defined for two parameters $\alpha, \beta \in (0, 1)$ by

$$\textsc{Bilinear}(x, y) := |y|(|x| - \beta n) - \alpha n|x|, \tag{3}$$

where we recall that for any bitstring $z \in \{0, 1\}^n$, $|z| := \sum_{i=1}^{n} z_i$ denotes the number of 1-bits in z. The function is illustrated in Figure 1 (left). Extended to the real domain, it is clear that the function is concave-convex, because $f(x) = g(x, y)$ is concave (linear) for all y, and $h(y) = g(x, y)$ is convex (linear) for all x. The gradient of the function is $\nabla g = (|y| - \alpha n, |x| - \beta n)$. Clearly, we have $\nabla g = 0$ when $|x| = \beta n$ and $|y| = \alpha n$.

Assuming that the prey (in $\mathcal{Y}$) always responds with an optimal decision for every $x \in X$, the predator is faced with the unimodal function f below which has maximum when $|x| = \beta n$.

$$f(x) := \min_{y \in \{0,1\}^n} g(x, y) = \begin{cases} |x|(1 - \alpha n) - \beta n & \text{if } |x| \leq \beta n \\ -\alpha n|x| & \text{if } |x| > \beta n. \end{cases}$$

We now characterise the dominated solutions wrt Bilinear.

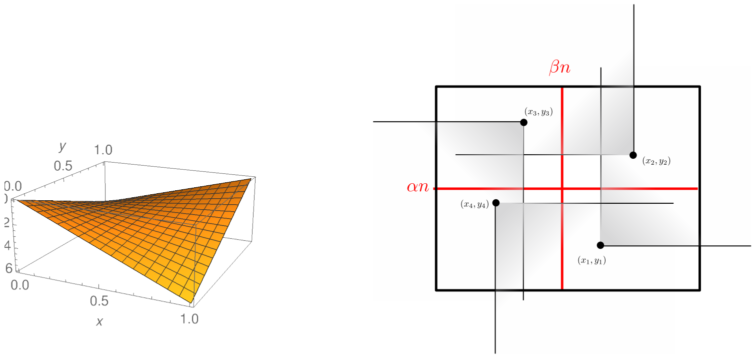

Figure 1: Left: Bilinear for $\alpha = 0.4$ and $\beta = 0.6$. Right: Dominance relationships in Bilinear.

Lemma 4. *Let* g :=Bilinear. *For all pairs* $(x_1, y_1), (x_2, y_2) \in \mathcal{X} \times \mathcal{Y}$, $(x_1, y_1) \succeq_g (x_2, y_2)$ *if and only if*

$$\begin{aligned} |y_2|(|x_1| - \beta n) &\geq |y_1|(|x_1| - \beta n) \quad \wedge \\ |x_1|(|y_1| - \alpha n) &\geq |x_2|(|y_1| - \alpha n). \end{aligned}$$

Proof. The proof follows from the definition of $\succeq_g$ and g:

$$\begin{aligned} & g(x_1, y_2) \geq g(x_1, y_1) \\ \iff \quad & |x_1||y_2| - \alpha n|x_1| - \beta n|y_2| \geq |x_1||y_1| - \alpha n|x_1| - \beta n|y_1| \\ \iff \quad & |y_2|(|x_1| - \beta n) \geq |y_1|(|x_1| - \beta n). \end{aligned}$$

The second part follows analogously from $g(x_1, y_1) \geq g(x_2, y_1)$. □

Figure 1 (right) illustrates Lemma 4, where the x-axis and y-axis correspond to the number of 1-bits in the predator x, respectively the number of 1-bits in the prey y. The figure contains four pairs, where the shaded area corresponds to the parts dominated by that pair: The pair (x_1, y_1) dominates (x_2, y_2), the pair (x_2, y_2) dominates (x_3, y_3), the pair (x_3, y_3) dominates (x_4, y_4), and the pair (x_4, y_4) dominates (x_1, y_1). This illustrates that the dominance-relation is intransitive. Lemma 5 states this and other properties of $\succeq_g$.

Lemma 5. *The relation* $\succeq_g$ *is reflexive, antisymmetric, and intransitive for* $g =$ Bilinear.

Proof. Reflexivity follows directly from the definition. Assume that $(x_1, y_1) \succeq_g (x_2, y_2)$ and $(x_1, y_1) \neq (x_2, y_2)$. Then, either $g(x_1, y_2) > g(x_1, y_2)$, or $g(x_1, y_1) > g(x_2, y_1)$, or both. Hence, $(x_2, y_2) \not\succeq_g (x_1, y_1)$, which proves that the relation is antisymmetric.

To prove intransitivity, it can be shown for any $\varepsilon > 0$, that $p_1 \succeq_g p_2 \succeq_g p_3 \succeq_g p_2 \succeq_g p_1$ where

$$p_1 = (\beta + \varepsilon, \alpha - 2\varepsilon) \qquad p_2 = (\beta - 2\varepsilon, \alpha - \varepsilon)$$
$$p_3 = (\beta - \varepsilon, \alpha + 2\varepsilon) \qquad p_4 = (\beta + 2\varepsilon, \alpha + \varepsilon). \qquad \square$$

We will frequently use the following simple lemma, which follows from the dominance relation and the definition of Bilinear.

Lemma 6. *For* Bilinear*, consider two samples* $(x_1, y_1), (x_2, y_2) \sim \text{Unif}(P \times Q)$*. Then the following conditional probabilities hold.*

$$\Pr\left((x_1, y_1) \succeq (x_2, y_2) \mid y_1 \leq y_2 \wedge x_1 > \beta n \wedge x_2 > \beta n\right) \geq 1/2$$
$$\Pr\left((x_1, y_1) \succeq (x_2, y_2) \mid y_1 \geq y_2 \wedge x_1 < \beta n \wedge x_2 < \beta n\right) \geq 1/2$$
$$\Pr\left((x_1, y_1) \succeq (x_2, y_2) \mid x_1 \geq x_2 \wedge y_1 > \alpha n \wedge y_2 > \alpha n\right) \geq 1/2$$
$$\Pr\left((x_1, y_1) \succeq (x_2, y_2) \mid x_1 \leq x_2 \wedge y_1 < \alpha n \wedge y_2 < \alpha n\right) \geq 1/2.$$

Proof. All the statements can be proved analogously, so we only show the first statement. If $y_1 \leq y_2$ and $x_1 > \beta n$, $x_2 > \beta n$, then by Lemma 4, $(x_1, y_1) \succeq (x_2, y_2)$ if and only if $x_1 \leq x_2$.

Since x_1 and x_2 are independent samples from the same (conditional) distribution, it follows that

$$1 \geq \Pr(x_1 > x_2) + \Pr(x_1 < x_2) = 2\Pr(x_1 > x_1) \tag{4}$$

Hence, we get $\Pr(x_1 \leq x_2) = 1 - \Pr(x_1 > x_2) \geq 1 - 1/2 = 1/2$. $\square$

5 A co-Evolutionary Algorithm for Maximin Optimisation

We now introduce a co-evolutionary algorithm for maximin optimisation (see Algorithm 2).

The predator and prey populations of size λ each are initialised uniformly at random in lines 1-3. Lines 6-17 describe how each pair of predator and prey are produced, first by selecting a predator-prey pair from the population, then applying mutation. In particular, the algorithm selects uniformly at random two predators x_1, x_2 and two prey y_1, y_2 in lines 7-8. The first pair (x_1, y_1) is *selected* if it dominates the second pair (x_2, y_2), otherwise the second pair is selected. The selected predator and prey are mutated by standard bitwise mutation in lines 14-15, i.e., each bit flips independently with probability χ/n (see Section C3.2.1 in Back et al. (1997)). The algorithm is a special case of the co-evolutionary framework in Section 2, where line 3 in Algorithm 1 corresponds to lines 6-17 in Algorithm 2.

Next, we will analyse the runtime of PD-CoEA on Bilinear using Theorem 3. For an arbitrary constant $\varepsilon > 0$, we will restrict the analysis to the

Algorithm 2 Pairwise Dominance CoEA (PD-CoEA)

```
Require: Min-max-objective function $g : \{0,1\}^n \times \{0,1\}^n \to \mathbb{R}$.
Require: Population size $\lambda \in \mathbb{N}$ and mutation rate $\chi \in (0, n]$
for $i \in [\lambda]$ do
    Sample $P_0(i) \sim \mathrm{Unif}(\{0,1\}^n)$
    Sample $Q_0(i) \sim \mathrm{Unif}(\{0,1\}^n)$
end for
for $t \in \mathbb{N}$ until termination criterion met do
    for $i \in [\lambda]$ do
        Sample $(x_1, y_1) \sim \mathrm{Unif}(P_t \times Q_t)$
        Sample $(x_2, y_2) \sim \mathrm{Unif}(P_t \times Q_t)$
        if $(x_1, y_1) \succeq_g (x_2, y_2)$ then
            $(x, y) := (x_1, y_1)$
        else
            $(x, y) := (x_2, y_2)$
        end if
        Obtain $x'$ by flipping each bit in $x$ with prob. $\chi/n$.
        Obtain $y'$ by flipping each bit in $y$ with prob. $\chi/n$.
        Set $P_{t+1}(i) := x'$ and $Q_{t+1}(i) := y'$.
    end for
end for
```

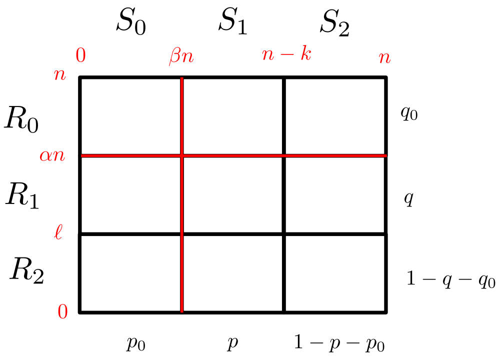

Figure 2: Partitioning of search space $\mathcal{X} \times \mathcal{Y}$ of Bilinear.

case where $\alpha - \varepsilon > 4/5$, and $\beta < \varepsilon$. Our goal is to estimate the time until the algorithm reaches within an ε-factor of the maximin-optimal point $(\beta n, \alpha n)$.

In this setting, the behaviour of the algorithm can be described intuitively as follows. The population dynamics will have two distinct phases. In Phase 1, most prey have less than αn 1-bits, while most predators have more than βn 1-bits. During this phase, predators and prey will decrease the number of 1-bits. In Phase 2, a sufficient number of predators have less than βn 1-bits, and the number of 1-bits in the prey-population will start to increase. The population will then reach the ε-approximation described above.

From this intuition, we will now define a suitable sequence of levels. We will start by dividing the space $\mathcal{X} \times \mathcal{Y}$ into different regions, as shown in Figure 2. Again, the x-axis corresponds to the number of 1-bits in the predator, while the y-axis corresponds to the number of 1-bits in the prey.

For any $k \in [0, (1-\beta)n]$, we partition $\mathcal{X}$ into three sets

$$S_0 := \{x \in \mathcal{X} \mid 0 \leq |x| < \beta n\} \tag{5}$$
$$S_1(k) := \{x \in \mathcal{X} \mid \beta n \leq |x| < n - k\}\,, \text{ and} \tag{6}$$
$$S_2(k) := \{x \in \mathcal{X} \mid n - k \leq |x| \leq n\}\,. \tag{7}$$

Similarly, for any $\ell \in [0, \alpha n)$, we partition $\mathcal{Y}$ into three sets

$$R_0 := \{y \in \mathcal{Y} \mid \alpha n \leq |y| \leq n\} \tag{8}$$
$$R_1(\ell) := \{y \in \mathcal{Y} \mid \ell \leq |y| < \alpha n\}\,, \text{ and} \tag{9}$$
$$R_2(\ell) := \{y \in \mathcal{Y} \mid 0 \leq |y| < \ell\}\,. \tag{10}$$

For ease of notation, when the parameters k and ℓ are clear from the context, we will simply refer to these sets as S_0, S_1, S_2, R_0, R_1, and R_2. Given two populations P and Q, and $C \subseteq \mathcal{X} \times \mathcal{Y}$, define

$$p(C) := \Pr_{(x,y)\sim\mathtt{Unif}(P\times Q)}((x,y) \in C)$$
$$p_{\mathrm{sel}}(C) := \Pr_{(x,y)\sim\mathtt{select}(P\times Q)}((x,y) \in C)\,.$$

In the context of subsets of $\mathcal{X} \times \mathcal{Y}$, the set S_i refers to $S_i \times \mathcal{Y}$, and R_i refers to $\mathcal{X} \times R_i$. With the above definitions, we will introduce the following quantities which depend on k and ℓ:

$$p_0 := p(S_0) \qquad p(k) := p(S_1(k)) \qquad q_0 := p(R_0) \qquad q(\ell) := p(R_1(\ell))$$

During Phase 1, the typical behaviour is that only a small minority of the individuals in the Q-population belong to region R_0. In this phase, the algorithm "progresses" by decreasing the number of 1-bits in the P-population. In this phase, the number of 1-bits will decrease in the Q-population, however it will not be necessary to analyse this in detail. To capture this, we define the levels for Phase 1 for $j \in [0..(1-\beta)n]$ as $A_j^{(1)} := S_0 \cup S_1(j)$ and $B_j^{(1)} := R_2((\alpha - \varepsilon)n)$.

During Phase 2, the typical behaviour is that there is a sufficiently large number of P-individuals in region S_0, and the algorithm progresses by increasing the number of 1-bits in the Q-population. The number of 1-bits in the P-population will decrease or stay at 0. To capture this, we define the levels for Phase 2 for $j \in [0, (\alpha - \varepsilon)n]$ $A_j^{(2)} := S_0$ and $B_j^{(2)} := R_1(j)$.

The overall sequence of levels used for Theorem 3 becomes

$$(A_0^{(1)} \times B_0^{(1)}), \ldots, (A_{(1-\beta)n}^{(1)}, B_{(1-\beta)n}^{(1)}), \\ (A_0^{(2)} \times B_0^{(2)}), \ldots, (A_{(\alpha-\varepsilon)n}^{(2)}, B_{(\alpha-\varepsilon)n}^{(2)}),$$

The notion of "current level" from Theorem 3 together with the level-structure can be exploited to infer properties about the populations, as the following lemma demonstrates.

Lemma 7. *If the current level is $A_j^{(1)} \times B_j^{(1)}$, then $p_0 < \gamma_0/(1-q_0)$.*

Proof. Assume by contradiction that $p_0(1-q_0) \geq \gamma_0$. Note that by (10), it holds $R_2(0) = \emptyset$. Therefore, $1 - q(0) - q_0 = 0$ and $q(0) = 1 - q_0$. By the definitions of the levels in Phase 2 and (9),

$$\begin{aligned}\left|(P \times Q) \cap (A_0^{(2)} \times B_0^{(2)})\right| &= |(P \times Q) \cap (S_0 \times R_1(0))| \\ &= p_0 q(0) \lambda^2 = p_0(1-q_0)\lambda^2 \geq \gamma_0 \lambda^2,\end{aligned}$$

implying that the current level must be level $A_0^{(2)} \times B_0^{(2)}$ or a higher level in Phase 2, contradicting the assumption of the lemma. □

5.1 Ensuring Condition (G2) during Phase 1

The purpose of this section is to provide the building-blocks necessary to establish conditions (G2a) and (G2b) during Phase 1. The progress of the population during this phase will be jeopardised if there are too many Q-individuals in R_0. We will employ the negative drift theorem for populations Lehre (2010) to prove that it is unlikely that Q-individuals will drift via region R_1 to region R_0. This theorem applies to algorithms that can be described on the form of Algorithm 3 which makes few assumptions about the selection step. The Q-population in Algorithm 2 is a special case of Algorithm 3.

We now state the negative drift theorem.

Theorem 8 (Negative Drift Theorem for Populations Lehre (2010))**.** *Given Algorithm 3 on $\mathcal{Y} = \{0,1\}^n$ with population size $\lambda \in \text{poly}(n)$, and transition matrix p_{mut} corresponding to flipping each bit independently with probability χ/n. Let $a(n)$ and $b(n)$ be positive integers s.t. $b(n) \leq n/\chi$ and $d(n) := b(n) - a(n) = \omega(\ln n)$. For an $x^* \in \{0,1\}^n$, let $T(n)$ be the smallest $t \geq 0$, s.t. $\min_{j \in [\lambda]} H(P_t(j), x^*) \leq a(n)$. Let $R_t(i) := \sum_{j=1}^{\lambda} [I_t(j) = i]$. If there are constants $\alpha_0 \geq 1$ and $\delta > 0$ such that*

1) $\mathbb{E}[R_t(i) \mid a(n) < H(P_t(i), x^*) < b(n)] \leq \alpha_0$ *for all $i \in [\lambda]$*

Algorithm 3 Population Selection-Variation Algorithm Lehre (2010)

```
Require: Finite state space $\mathcal{Y}$.
Require: Transition matrix $p_{\mathrm{mut}}$ over $\mathcal{Y}$.
Require: Population size $\lambda \in \mathbb{N}$.
Require: Initial population $Q_0 \in \mathcal{Y}^\lambda$.
for $t = 0, 1, 2, \dots$ until the termination condition is met do
    for $i = 1$ to $\lambda$ do
        Choose $I_t(i) \in [\lambda]$, and set $x := Q_t(I_t(i))$.
        Sample $x' \sim p_{\mathrm{mut}}(x)$ and set $Q_{t+1}(i) := x'$.
    end for
end for
```

2) *$\psi := \ln(\alpha_0)/\chi + \delta < 1$, and*

3) $\frac{b(n)}{n} < \min\left\{\frac{1}{5}, \frac{1}{2} - \frac{1}{2}\sqrt{\psi(2-\psi)}\right\}$,

then $\Pr\left(T(n) \le e^{cd(n)}\right) \le e^{-\Omega(d(n))}$ *for some constant* $c > 0$.

To apply this theorem, the first step is to estimate the reproductive rate Lehre (2010) of Q-individuals in $R_0 \cup R_1$.

Lemma 9. *If there exist constants $\delta_1, \delta_2 \in (0,1)$ such that $q + q_0 \le 1 - \delta_1$, $p_0 < \sqrt{2(1-\delta_2)} - 1$, and $p_0 q = 0$, then there exists a constant $\delta \in (0,1)$ such that $p_{sel}(R_0 \cup R_1)/p(R_0 \cup R_1) < 1 - \delta$.*

Lemma 10. *If $p_0 = 0$ and $q_0 + q \le 1/3$, then no Q-individual in $Q \cap (R_0 \cup R_1)$ has reproductive rate higher than* 1.

Proof. Consider any individual $z \in Q \cap (R_0 \cup R_1)$. The probability of selecting this individual in a given iteration is less than

$$\begin{aligned}
&\Pr\left(y_1 = z \wedge y_2 \in R_0 \cup R_1\right) \Pr\left((x_1, y_1) \succeq (x_2, y_2) \mid y_1 = z \wedge y_2 \in R_0 \cup R_1\right) \\
&+ \Pr\left(y_2 = z \wedge y_1 \in R_0 \cup R_1\right) \Pr\left((x_1, y_1) \not\succeq (x_2, y_2) \mid y_2 = z \wedge y_1 \in R_0 \cup R_1\right) \\
&\quad + \Pr\left(y_2 = z \wedge y_1 \in R_2\right) \Pr\left((x_1, y_1) \not\succeq (x_2, y_2) \mid y_2 = z \wedge y_1 \in R_2\right) \\
&\qquad \le \frac{2}{\lambda}(q_0 + q) + (1 - q - q_0)/(2\lambda) = \frac{1}{2\lambda}\left(1 + 3(q + q_0)\right) \le \frac{1}{\lambda}.
\end{aligned}$$

Hence, within one generation of λ iterations, the expected number of times this individual is selected is at most 1. □

We now have the necessary ingredients to prove the required condition about the number of Q-individuals in R_0.

Lemma 11. *Assume that $\lambda \in \mathrm{poly}(n)$, and for two constants $\alpha, \varepsilon \in (0,1)$ with $\alpha - \varepsilon \ge 4/5$, the mutation rate is $\chi \le 1/(1 - \alpha + \varepsilon)$. Let T be as defined in*

Theorem 14. For any $\tau \leq e^{cn}$ where c is a sufficiently small constant, define $\tau_* := \min\{T/\lambda - 1, \tau\}$, then

$$\Pr\left(\bigvee_{t=0}^{\tau_*} (Q_t \cap R_0) \neq \emptyset\right) \leq \tau e^{-\Omega(n)} + \tau e^{-\Omega(\lambda)}.$$

Proof. Each individual in the initial population Q_0 is sampled uniformly at random, with $n/2 \leq (\alpha - \varepsilon)n/(1 + 3/5)$ expected number of 1-bits. Hence, by a Chernoff bound Motwani and Raghavan (1995) and a union bound, the probability that the initial population Q_0 intersects with $R_0 \cup R_1$ is no more than $\lambda e^{-\Omega(n)} = e^{-\Omega(n)}$.

We divide the remaining $t - 1$ generations into a random number of phases, where each phase lasts until $p_0 > 0$, and we assume that the phase begins with $q_0 = 0$.

If a phase begins with $p_0 > 0$, then the phase lasts one generation. Furthermore, it must hold that $q((\alpha - \varepsilon)n) = 0$, otherwise the product $P_t \times Q_t$ contains a pair in $S_0 \times R_1((\alpha - \varepsilon)n)$, i.e., an ε-approximate solution has been found, which contradicts that $t < T/\lambda$. If $q((\alpha - \varepsilon)n) = 0$, then all Q-individuals belong to region R_2. In order to obtain any Q-individual in region R_0, it is necessary that at least one of λ individuals mutates at least εn 0-bits, an event which holds with probability at most $\lambda \cdot \binom{n}{\varepsilon n} \left(\frac{\chi}{n}\right)^{\varepsilon n} \leq \lambda e^{-\Omega(n)} = e^{-\Omega(n)}$.

If a phase begins with $p_0 = 0$, then we will apply Theorem 8 to show that it is unlikely that any Q-individual will reach R_0 within e^{cn} generations, or the phase ends. We use the parameter $x^* := 1^n$, $a(n) := (1 - \alpha)n$, and $b(n) := (1 - \alpha + \varepsilon)n < n/\chi$. Hence, $d(n) := b(n) - a(n) = \varepsilon n = \omega(\ln(n))$.

We first bound the reproductive rate of Q-individuals in R_1. For any generation t, if $q_0 + q < (1 - \delta_2)$, then by Lemma 9, and a Chernoff bound, $|Q_{t+1} \cap R_0 \cup R_1| \leq (q_0 + q)\lambda$ with probability $1 - e^{-\Omega(\lambda)}$. By a union bound, this holds with probability $1 - te^{-\Omega(\lambda)}$ within the next t generations. Hence, by Lemma 10, the reproductive rate of any Q-individual within $R_0 \cup R_1$ is at most $\alpha_0 := 1$, and condition 1 of Theorem 8 is satisfied. Furthermore, $\psi := \ln(\alpha_0)/\chi + \delta = \delta' < 1$ for any $\delta' \in (0, 1)$ and $\chi > 0$, hence condition 2 is satisfied. Finally, condition 3 is satisfied as long as δ' is chosen sufficiently small. It follows by Theorem 8 that the probability that a Q-individual in R_0 is produced within a phase of length at most $\tau < e^{cn}$ is $e^{-\Omega(n)}$.

The lemma now follows by taking a union bound over the at most τ phases. □

We can now proceed to analyse Phase 1, assuming that $q_0 = 0$. For a lower bound and to simplify calculations, we pessimistically assume that the following event occurs with probability 0

$$(x_1, y_1) \in S_1 \times R_1 \cup R_2 \ \wedge \ (x_2, y_2) \in S_2 \cup R_0.$$

Lemma 12. *If there exist constants $\delta, \psi \in (0, 1)$ such that*

1) $p_0 \leq \sqrt{2(1 - \delta)} - 1$

2) $q_0 \leq \sqrt{2(1-\delta)} - 1$

3) $p_0 q = 0$

then if $(p_0 + p)(1 - q - q_0) \leq \psi$*, it holds that*

$$\varphi := \frac{p_{sel}(S_0 \cup S_1)}{p(S_0 \cup S_1)} \cdot \frac{p_{sel}(R_2)}{p(R_2)} \geq 1 + \delta(1 - \sqrt{\psi}),$$

otherwise, if $(p_0 + p)(1 - q - q_0) \geq \psi$*, then* $p_{sel}(S_0 \cup S_1) p_{sel}(R_2) \geq \psi$*.*

Proof. Given the assumptions, Lemma 23 and Lemma 24 imply

$$\varphi \geq (1 + \delta(1 - p - p_0))(1 + \delta(q + q_0)) \geq 1. \tag{11}$$

For the first statement, we consider two cases:

Case 1: If $p_0 + p < \sqrt{\psi}$, then by (11) and $q_0 + q \geq 0$, it follows $\varphi \geq (1 + \delta(1 - \sqrt{\psi})) \cdot 1$.

Case 2: If $p_0 + p \geq \sqrt{\psi}$, then by assumption $(1 - q - q_0) \leq \sqrt{\psi}$. By (11) and $1 - p - p_0 \geq 0$, it follows that $\varphi \geq 1 \cdot (1 + \delta(1 - \sqrt{\psi}))$.

For the second statement, (11) implies

$$\begin{aligned} p_{\mathrm{sel}}(S_0 \cup S_1) p_{\mathrm{sel}}(R_2) &= \varphi p(S_0 \cup S_1) p(R_2) \\ &= \varphi (p_0 + p)(1 - q - q_0) \geq 1 \cdot \psi. \end{aligned}$$ □

5.2 Ensuring Condition (G2) during Phase 2

We now proceed to analyse Phase 2.

Corollary 13. *For any constant* $\delta \in (0,1)$*, if* $\gamma_0 < 1 - \delta$ *and* $0 \leq q_0 < \delta/1200$ *and* $p_0 \in (1/3, 1]$*, then there exists a constant* $\delta' > 0$ *such that*

$$\frac{p_{sel}(S_0)}{p(S_0)} \frac{p_{sel}(R_1)}{p(R_1)} > 1 + \delta'.$$

Proof. We distinguish between two cases. If $p_0 \in (1/3, 1 - \delta/10)$, we apply Lemma 19. The conditions of Lemma 19 hold for the parameter $\delta_1 := \delta/10$, and the statement follows for $\delta' = \delta_1'$. If $p_0 \in (1 - \delta/10, 1]$, the statement follows immediately from Lemma 19 for the parameter $\delta' = 23\delta/300$. □

5.3 Main Result

We now obtain the main result: Algorithm 2 can efficiently locate an ε-approximate solution to an instance of Bilinear.

Theorem 14. *Assume that for a sufficiently large constant c, it holds* $c \log(n) \leq \lambda \in \mathrm{poly}(n)$*. Let* $\alpha, \beta, \varepsilon \in (0,1)$ *be three constants where* $\alpha - \varepsilon \geq 4/5$*. Define* $T := \min\{\lambda t \mid (P_t \times Q_t) \cap S_0 \times R_1((\alpha - \varepsilon))n\}$ *where* P_t *and* Q_t *are the populations of Algorithm 2 applied to* Bilinear $_{\alpha,\beta}$*. Then if the mutation rate* χ *is a sufficiently small constant, and at most* $1/(1 - \alpha + \varepsilon)$*, there exists a constant* c_0 *such that for all* $r \in \mathrm{poly}(n)$*, it holds* $\Pr\left(T > c_0 r \lambda^3 n\right) \leq (1/r)(1 + o(1))$*.*

6 A Co-Evolutionary Error Threshold

The previous section presented a scenario where Algorithm 2 obtains an approximate solution efficiently. We now present a general scenario where the algorithm is inefficient. In particular, we show that there exists a critical mutation rate above which the algorithm fails on any problem, as long as the problem does not have too many optima. The critical mutation rate is called the "error threshold" of the algorithm Ochoa (2006); Lehre (2010). As far as the author is aware, this is the first time an error threshold has been identified in co-evolution.

Theorem 15. *There exists a constant $c > 0$ such that the following holds. If A and B are subsets of $\{0,1\}^n$ with $\min\{|A|,|B|\} \leq e^{cn}$, and Algorithm 2 is executed with population size $\lambda \in \mathrm{poly}(n)$ and constant mutation rate $\chi > \ln(2)/(1-2\delta)$ for any constant $\delta \in (0,1/2)$, then there exists a constant c' such that $\Pr\left(T_{A\times B} < e^{c'n}\right) = e^{-\Omega(n)}$.*

Proof. Without loss of generality, assume that $|B| \leq |A|$. For a lower bound on $T_{A\times B}$, it suffices to compute a lower bound on the time until the Q-population contains an element in B.

For any $y \in B$, we will apply Theorem 8 to bound $T_y := \min\{t \mid H(Q_t, y) \leq 0\}$, i.e., the time until the Q population contains y. Define $a(n) := 0$ and $b(n) := n\min\{1/5, 1/2 - (1/2)\sqrt{1-\delta^2}, 1/\chi\}$. Since δ is a constant, it follows that $d(n) = b(n) - a(n) = \omega(\ln n)$. Furthermore, by definition, $b(n) \leq n/\chi$.

We now show that condition 1 of Theorem 8 holds for $\alpha_0 := 2$. For any individual $u \in \mathcal{Y}$, the probability that the individual is selected in lines 7-12 is at most $1 - \Pr(y_1 \neq u \wedge y_2 \neq u) = 1 - (1-1/\lambda)^2 = (1/\lambda)(2-1/\lambda)$. Thus within the λ iterations, individual u is selected less than 2 times in expectation. This proves condition 1.

Condition 2 is satisfied because by the assumption on the mutation rate, $\psi := \ln(\alpha_0)/\chi + \delta \leq 1 - \delta < 1$. Finally, condition 3 trivially holds because $b(n) \leq n/5$ and $1/2 - \sqrt{\psi(2-\psi)}/2 \leq 1/2 - \sqrt{1-\delta^2}/2 \leq b(n)/n$.

All conditions are satisfied, and Theorem 8 imply that for some constant c', $\Pr\left(T_{y^*} < e^{c'n}\right) = e^{-\Omega(n)}$. Taking a union bound over all elements in B, we get for sufficiently small c $\Pr\left(T_{A\times B} < e^{c'n}\right) \leq \Pr\left(T_{B\times\mathcal{Y}} < e^{c'n}\right) \leq \sum_{y\in B}\Pr\left(T_y < e^{c'n}\right) \leq e^{cn}\cdot e^{-\Omega(n)} = e^{-\Omega(n)}$. □

7 Conclusion

Co-evolutionary algorithms have gained wide-spread interest, with a number of exciting applications. However, their population dynamics tend to be significantly more complex than in standard evolutionary algorithms. A number of pathological behaviours are reported in the literature, preventing the potential of these algorithms. There has been a long-standing goal to develop a rigorous theory for co-evolution which can explain when they are efficient. A major obstacle for such a theory is to reason about the complex interactions that occur between multiple populations.

This paper provides the first step in developing runtime analysis for population-based, competitive co-evolutionary algorithms. A generic mathematical framework covering a wide range of CoEAs is presented, along with an analytical tool to derive upper bounds on their expected runtimes. To illustrate the approach, we define a new co-evolutionary algorithm PD-CoEA and analyse its runtime on a bilinear maximin-optimisation problem Bilinear. For some problem instances, the algorithm obtains a solution within arbitrary constant approximation ratio to the optimum within polynomial time $O(r\lambda^3 n)$ with probability $1-(1/r)(1+o(1))$ for all $r \in \text{poly}(n)$, assuming population size $\lambda \in \Omega(\log n) \cap \text{poly}(n)$ and sufficiently small (but constant) mutation rate. Additionally, we present a setting where PD-CoAE is inefficient. In particular, if the mutation rate is too high, the algorithm needs with overwhelmingly high probability exponential time to reach any fixed solution. This constitutes a co-evolutionary "error threshold".

Future work should consider broader classes of problems, as well as other co-evolutionary algorithms.

Acknowledgements

Lehre was supported by a Turing AI Fellowship (EPSRC grant ref EP/V025562/1).

References

Abdullah Al-Dujaili, Shashank Srikant, Erik Hemberg, and Una-May O'Reilly. 2019. On the application of Danskin's theorem to derivative-free minimax problems. *AIP Conference Proceedings* 2070, 1 (Feb. 2019), 020026. https://doi.org/10.1063/1.5089993 Publisher: American Institute of Physics.

Andrea Arcuri and Xin Yao. 2008. A novel co-evolutionary approach to automatic software bug fixing. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. 162–168. https://doi.org/10.1109/CEC.2008.4630793 ISSN: 1941-0026.

Thomas Back, David B. Fogel, and Zbigniew Michalewicz. 1997. *Handbook of Evolutionary Computation* (1st ed.). IOP Publishing Ltd., GBR.

Dogan Corus, Duc-Cuong Dang, Anton V. Eremeev, and Per Kristian Lehre. 2018. Level-Based Analysis of Genetic Algorithms and Other Search Processes. *IEEE Transactions on Evolutionary Computation* 22, 5 (Oct. 2018), 707–719. https://doi.org/10.1109/TEVC.2017.2753538

Duc-Cuong Dang and Per Kristian Lehre. 2016. Runtime Analysis of Non-elitist Populations: From Classical Optimisation to Partial Information.

Algorithmica 75, 3 (July 2016), 428–461. https://doi.org/10.1007/s00453-015-0103-x

Benjamin Doerr and Frank Neumann (Eds.). 2020. *Theory of Evolutionary Computation*. Springer.

Stefan Droste, Thomas Jansen, and Ingo Wegener. 2006. Upper and Lower Bounds for Randomized Search Heuristics in Black-Box Optimization. *Theory of Computing Systems* 39, 4 (July 2006), 525–544. https://doi.org/10.1007/s00224-004-1177-z

John Fearnley and Rahul Savani. 2016. Finding Approximate Nash Equilibria of Bimatrix Games via Payoff Queries. *ACM Trans. on Economics and Computation* 4, 4 (Aug. 2016), 1–19. https://doi.org/10.1145/2956579

Sevan G Ficici. 2004. *Solution Concepts in Coevolutionary Algorithms*. Ph. D. Dissertation. Brandeis University.

W. Daniel Hillis. 1990. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena* 42, 1 (June 1990), 228–234. https://doi.org/10.1016/0167-2789(90)90076-2

Thomas Jansen and R. Paul Wiegand. 2004. The Cooperative Coevolutionary (1+1) EA. *Evolutionary Computation* 12, 4 (Dec. 2004), 405–434. https://doi.org/10.1162/1063656043138905

Mikkel T. Jensen. 2004. A New Look at Solving Minimax Problems with Coevolutionary Genetic Algorithms. In *Metaheuristics: Computer Decision-Making*, Mauricio G. C. Resende and Jorge Pinho de Sousa (Eds.). Springer US, Boston, MA, 369–384. https://doi.org/10.1007/978-1-4757-4137-7_17

Per Kristian Lehre. 2010. Negative Drift in Populations. In *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN 2010) (LNCS, Vol. 6238)*. Springer Berlin / Heidelberg, 244–253. https://doi.org/10.1007/978-3-642-15844-5_25

Per Kristian Lehre. 2011. Fitness-levels for non-elitist populations. *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11* (2011), 2075. https://doi.org/10.1145/2001576.2001855

Atsuhiro Miyagi, Kazuto Fukuchi, Jun Sakuma, and Youhei Akimoto. 2021. Adaptive scenario subset selection for min-max black-box continuous optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 697–705. https://doi.org/10.1145/3449639.3459291

Rajeev Motwani and Prabhakar Raghavan. 1995. *Randomized Algorithms*. Cambridge University Press.

Gabriela Ochoa. 2006. Error Thresholds in Genetic Algorithms. *Evolutionary Computation* 14, 2 (June 2006), 157–182. https://doi.org/10.1162/evco.2006.14.2.157

Una-May O'Reilly, Jamal Toutouh, Marcos Pertierra, Daniel Prado Sanchez, Dennis Garcia, Anthony Erb Luogo, Jonathan Kelly, and Erik Hemberg. 2020. Adversarial genetic programming for cyber security: a rising application domain where GP matters. *Genetic Programming and Evolvable Machines* 21, 1-2 (June 2020), 219–250. https://doi.org/10.1007/s10710-020-09389-y

Jordan B. Pollack, Hod Lipson, Gregory Hornby, and Pablo Funes. 2001. Three Generations of Automatically Designed Robots. *Artificial Life* 7, 3 (July 2001), 215–223. https://doi.org/10.1162/106454601753238627

Elena Popovici, Anthony Bucci, R. Paul Wiegand, and Edwin D. De Jong. 2012. Coevolutionary Principles. In *Handbook of Natural Computing*, Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 987–1033. https://doi.org/10.1007/978-3-540-92910-9_31

Mitchell A. Potter and Kenneth A. De Jong. 2000. Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation* 8, 1 (March 2000), 1–29. https://doi.org/10.1162/106365600568086

Richard A. Watson and Jordan B. Pollack. 2001. Coevolutionary Dynamics in a Minimal Substrate. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO'01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 702–709. http://dl.acm.org/citation.cfm?id=2955239.2955343 event-place: San Francisco, California.

Ingo Wegener. 2002. Methods for the Analysis of Evolutionary Algorithms on Pseudo-Boolean Functions. In *Evolutionary Optimization*, Ruhul Sarker, Masoud Mohammadian, and Xin Yao (Eds.). Springer US, Boston, MA, 349–369. https://doi.org/10.1007/0-306-48041-7_14

A Proof of the Level-based Theorem

This section provides the proof of the level-based theorem. The proof follows closely the proof of the original level-based theorem, however there are some notable differences, particularly in the assumptions about the underlying stochastic process and the choice of the "level-functions". For ease of comparison, we have kept the proof identical to the classical proof where possible.

Definition 3 (Corus et al. (2018))**.** *A function* $g : (\{0\} \cup [\lambda^2]) \times [m] \to \mathbb{R}$ *is called a* level function *if the following three conditions hold*

1. $\forall x \in \{0\} \cup [\lambda^2], \forall y \in [m-1] : \quad g(x, y) \geq g(x, y+1),$
2. $\forall x \in \{0\} \cup [\lambda^2 - 1], \forall y \in [m] : \quad g(x, y) \geq g(x+1, y),$
3. $\forall y \in [m-1] : \quad g(\lambda^2, y) \geq g(0, y+1).$

It follows directly from the definition that the set of level-functions is closed under addition.

Lemma 16 (Corus et al. (2018))**.** *If* $Y_{t+1} \geq Y_t$, *then for any level function* g

$$g\left(X_{t+1}^{(Y_{t+1}+1)}, Y_{t+1}\right) \leq g\left(X_{t+1}^{(Y_t+1)}, Y_t\right).$$

Proof. The statement is trivially true when $Y_t = Y_{t+1}$. On the other hand, if $Y_{t+1} \geq Y_t + 1$, then the conditions in Definition 3 imply

$$\begin{aligned} g\left(X_{t+1}^{(Y_{t+1}+1)}, Y_{t+1}\right) &\leq g\left(0, Y_{t+1}\right) \leq g\left(0, Y_t + 1\right) \\ &\leq g\left(\lambda^2, Y_t\right) \leq g\left(X_{t+1}^{(Y_t+1)}, Y_t\right). \end{aligned}$$

□

Proof of Theorem 3. We apply Theorem 26 (the additive drift theorem) with respect to the parameter $a = 0$ and the process $Z_t := g\left(X_t^{(Y_t+1)}, Y_t\right)$, where g is a level-function, and $(Y_t)_{t\in\mathbb{N}}$ and $(X_t^{(j)})_{t\in\mathbb{N}}$ for $j \in [m]$ are stochastic processes, which will be defined later. $(\mathscr{F}_t)_{t\in\mathbb{N}}$ is the filtration induced by the populations $(P_t)_{t\in\mathbb{N}}$ and $(Q_t)_{t\in\mathbb{N}}$.

We will assume w.l.o.g. that conditions (G2a) and (G2b) are also satisfied for $j = m-1$, for the following reason. Given Algorithm 1 with a certain mapping $\mathcal{D}$, consider Algorithm 1 with a different mapping $\mathcal{D}'(P, Q)$: If $(P \times Q) \cap (A_m \times B_m) = \emptyset$, then $\mathcal{D}'(P, Q) = \mathcal{D}(P, Q)$; otherwise $\mathcal{D}'(P, Q)$ assigns probability mass 1 to some pair (x, y) of $P \times Q$ that is in A_m, e.g., to the first one among such elements. Note that $\mathcal{D}'$ meets conditions (G1) and (G2). Moreover, (G2a) and (G2b) hold for $j = m-1$. For the sequence of populations $P'_0, P'_1, \ldots$ and $Q'_0, Q'_1, \ldots$ of Algorithm 1 with mapping $\mathcal{D}'$, we can put $T' := \min\{\lambda t \mid (P'_t \times Q'_t) \cap (A_m \times B_m) \neq \emptyset\}$. Executions of the original algorithm and the modified one before generation T'/λ are identical. On generation T'/λ both algorithms place elements of A_m into the populations for the first time. Thus, T' and T are equal in every realisation and their expectations are equal.

For any level $j \in [m]$ and time $t \geq 0$, let the random variable $X_t^{(j)} := |(P_t \times Q_t) \cap (A_j \times B_j)|$ denote the number of pairs in level $A_j \times B_j$ at time t. The *current level* Y_t of the algorithm at time t is defined as $Y_t := \max \left\{ j \in [m] \;\middle|\; X_t^{(j)} \geq \gamma_0 \lambda^2 \right\}$. Note that $(X_t^{(j)})_{t\in\mathbb{N}}$ and $(Y_t)_{t\in\mathbb{N}}$ are adapted to the filtration $(\mathscr{F}_t)_{t\in\mathbb{N}}$ because they are defined in terms of the populations $(P_t)_{t\in\mathbb{N}}$ and $(Q_t)_{t\in\mathbb{N}}$.

When $Y_t < m$, there exists a unique $\gamma \in [0, \gamma_0)$ such that

$$X_t^{(Y_t+1)} = |(P_t \times Q_t) \cap (A_{Y_t+1} \times B_{Y_t+1})| = \gamma\lambda^2, \text{ and} \tag{12}$$

$$X_t^{(Y_t)} = |(P_t \times Q_t) \cap (A_{Y_t} \times B_{Y_t})| \geq \gamma_0\lambda^2. \tag{13}$$

Finally, we define the process $(Z_t)_{t\in\mathbb{N}}$ as $Z_t := 0$ if $Y_t = m$, and otherwise, if $Y_t < m$, we let

$$Z_t := g\left(X_t^{(Y_t+1)}, Y_t\right),$$

where for all $k \in [\lambda^2]$, and for all $j \in [m-1]$, $g(k,j) := g_1(k,j) + g_2(k,j)$ and

$$g_1(k,j) := \frac{\eta}{1+\eta} \cdot ((m-j)\lambda^2 - k)$$

$$g_2(k,j) := \varphi \cdot \left(\frac{e^{-\eta k}}{q_j} + \sum_{i=j+1}^{m-1} \frac{1}{q_i} \right),$$

where the parameters $\eta, \varphi \in (0,1)$ will be specified later, and for $j \in [m-1]$, $q_j := \lambda z_j / (4 + \lambda z_j)$.

Both functions have partial derivatives $\frac{\partial g_i}{\partial k} < 0$ and $\frac{\partial g_i}{\partial j} < 0$, hence they satisfy properties 1 and 2 of Definition 3. They also satisfy property 3 because for all $j \in [m-1]$

$$g_1(\lambda^2, j) = \frac{\eta}{1+\eta}((m-j)\lambda^2 - \lambda^2) = g_1(0, j+1)$$

$$g_2(\lambda^2, j) > \sum_{i=j+1}^{m-1} \frac{\varphi}{q_i} = g_2(0, j+1).$$

Therefore g_1 and g_2 are level-functions, and thus also their linear combination g is a level function.

Due to properties 1 and 2 of level-functions (see Definition 3), it holds for all $k \in [0..\lambda^2]$ and $j \in [m-1]$

$$0 \leq g(k,j) \leq g(0,1) < \frac{\eta m \lambda^2}{1+\eta} + \sum_{i=1}^{m-1} \frac{\varphi}{q_i} \tag{14}$$

$$< \frac{\eta m \lambda^2}{1+\eta} + \varphi \sum_{i=1}^{m-1} \frac{4+\lambda z_i}{\lambda z_i} \tag{15}$$

$$< m\left(2\eta\lambda^2 + \frac{4\varphi}{\lambda z_*}\right). \tag{16}$$

Hence, we have $0 \leq Z_t < g(0,1) < \infty$ for all $t \in \mathbb{N}$ which implies that condition 2 of the drift theorem is satisfied.

The drift of the process at time t is $\mathbb{E}_t\left[\Delta_{t+1}\right]$, where

$$\Delta_{t+1} := g\left(X_t^{(Y_t+1)}, Y_t\right) - g\left(X_{t+1}^{(Y_{t+1}+1)}, Y_{t+1}\right).$$

We bound the drift by the law of total probability as

$$\begin{aligned}\mathbb{E}_t\left[\Delta_{t+1}\right] = (1 - \Pr{}_t\left(Y_{t+1} < Y_t\right))\mathbb{E}_t\left[\Delta_{t+1} \mid Y_{t+1} \geq Y_t\right] \\ + \Pr{}_t\left(Y_{t+1} < Y_t\right)\mathbb{E}_t\left[\Delta_{t+1} \mid Y_{t+1} < Y_t\right].\end{aligned} \tag{17}$$

The event $Y_{t+1} < Y_t$ holds if and only if $X_{t+1}^{(Y_t)} < \gamma_0\lambda^2$, which by Lemma 1 statement 3, condition (G2b), and condition (G3) with a sufficiently large constant c' such that $c'\lambda > (2/c)\log(4m/z_*)$ is upper bounded by

$$\Pr{}_t\left(Y_{t+1} < Y_t\right) = \Pr{}_t\left(X_{t+1}^{(Y_t)} < \gamma_0\lambda^2\right) \tag{18}$$

$$< e^{-c\lambda} < e^{-(c/2)\lambda}e^{-(c/2)\lambda} \qquad < \frac{48}{(c\lambda)^3} \cdot \frac{z_*}{4m}, \tag{19}$$

where the last inequality uses $e^x > x^3/3!$ from the Maclaurin series of the exponential function. Given the low probability of the event $Y_{t+1} < Y_t$, it suffices to use the pessimistic bound (16)

$$\mathbb{E}_t\left[\Delta_{t+1} \mid Y_{t+1} < Y_t\right] \geq -g(0,1) \tag{20}$$

If $Y_{t+1} \geq Y_t$, we can apply Lemma 16

$$\begin{aligned}&\mathbb{E}_t\left[\Delta_{t+1} \mid Y_{t+1} \geq Y_t\right] \\ &\quad \geq \mathbb{E}_t\left[g\left(X_t^{(Y_t+1)}, Y_t\right) - g\left(X_{t+1}^{(Y_t+1)}, Y_t\right) \mid Y_{t+1} \geq Y_t\right].\end{aligned}$$

If $X_t^{(Y_t+1)} = 0$, then $X_t^{(Y_t+1)} \leq X_{t+1}^{(Y_t+1)}$ and

$$\mathbb{E}_t\left[g_1\left(X_t^{(Y_t+1)}, Y_t\right) - g_1\left(X_{t+1}^{(Y_t+1)}, Y_t\right) \mid Y_{t+1} \geq Y_t\right] \geq 0,$$

because the function g_1 satisfies property 2 in Definition 3. Furthermore, we have the lower bound

$$\begin{aligned}\mathbb{E}_t & \left[g_2\left(X_t^{(Y_t+1)}, Y_t\right) - g_2\left(X_{t+1}^{(Y_t+1)}, Y_t\right) \mid Y_{t+1} \geq Y_t\right] \\ & > \Pr{}_t\left(X_{t+1}^{(Y_t+1)} \geq 1\right)\left(g_2\left(0, Y_t\right) - g_2\left(1, Y_t\right)\right) \geq \frac{\eta\varphi}{1+\eta}.\end{aligned}$$

where the last inequality follows because

$$\begin{aligned}\Pr{}_t\left(X_{t+1}^{(Y_t+1)} \geq 1\right) &= \Pr{}_t\left(\left(P_{t+1} \times Q_{t+1}\right) \cap \left(A_{Y_t+1} \times B_{Y_t+1}\right) \neq \emptyset\right) \\ &\geq q_{Y_t},\end{aligned}$$

due to condition (G1) and Lemma 2, and

$$g_2\left(0, Y_t\right) - g_2\left(1, Y_t\right) = (\varphi/q_{Y_t})(1 - e^{-\eta}) \geq \frac{\varphi\eta}{(1+\eta)q_{Y_t}}$$

In the other case, where $X_t^{(Y_t+1)} = \gamma\lambda^2 \geq 1$, Lemma 1 and condition (G2a) imply for $\varphi := \delta(1-\delta')$ for an arbitrary constant $\delta' \in (0,1)$,

$$\begin{aligned}\mathbb{E}_t & \left[g_1\left(X_t^{(Y_t+1)}, Y_t\right) - g_1\left(X_{t+1}^{(Y_t+1)}, Y_t\right) \mid Y_{t+1} \geq Y_t\right] \\ &= \frac{\eta}{1+\eta}\mathbb{E}_t\left[X_{t+1}^{(Y_t+1)} \mid Y_{t+1} \geq Y_t\right] - \frac{\eta}{1+\eta}X_t^{(Y_t+1)} \\ &\geq \frac{\eta}{1+\eta}(\lambda(\lambda-1)(1+\delta)\gamma - \gamma\lambda^2) > \frac{\eta}{1+\eta}\delta(1-\delta') = \frac{\eta\varphi}{1+\eta}, \quad (21)\end{aligned}$$

where the last inequality is obtained by choosing the minimal value $\gamma = 1/\lambda^2$. For the function g_2, we get

$$\begin{aligned}\mathbb{E}_t & \left[g_2\left(X_t^{(Y_t+1)}, Y_t\right) - g_2\left(X_{t+1}^{(Y_t+1)}, Y_t\right) \mid Y_{t+1} \geq Y_t\right] = \\ & \frac{\varphi}{q_{Y_t}}\left(e^{-\eta X_t^{(Y_t+1)}} - \mathbb{E}_t\left[e^{-\eta X_{t+1}^{(Y_t+1)}}\right]\right) > 0,\end{aligned}$$

where the last inequality is due to statement 2 of Lemma 1 for the parameter $\eta := (1-(1+\delta)^{-1/2})/\lambda$.

Taking into account all cases, we have

$$\mathbb{E}_t\left[\Delta_{t+1} \mid Y_{t+1} \geq Y_t\right] \geq \frac{\eta\varphi}{1+\eta}. \quad (22)$$

We now have bounds for all the quantities in (17) with (19), (20), and (22), and we get

$$\begin{aligned}\mathbb{E}_t\left[\Delta_{t+1}\right] &= (1 - \Pr{}_t\left(Y_{t+1} < Y_t\right))\mathbb{E}_t\left[\Delta_{t+1} \mid Y_{t+1} \geq Y_t\right] \\ &\quad + \Pr{}_t\left(Y_{t+1} < Y_t\right)\mathbb{E}_t\left[\Delta_{t+1} \mid Y_{t+1} < Y_t\right] \\ &\geq \frac{\eta\varphi}{1+\eta} - \frac{48}{(c\lambda)^3}\frac{z_*}{4m}\left(m2\eta\lambda^2 + \frac{4m\varphi}{\lambda z_*} + \frac{\eta\varphi}{1+\eta}\right) \\ &> \frac{\eta\varphi(1-\delta'')}{1+\eta}\end{aligned}$$

for an arbitrary small constant δ'' as long as m and λ are sufficiently large.

We now verify condition 3 of Theorem 26, i.e., that T has finite expectation. Let $p_* := \min\{(1+\delta)(1/\lambda^2), z_*\} > 0$, and note by conditions (G1) and (G2a) that the current level increases by at least one with probability $\Pr_t(Y_{t+1} > Y_t) \geq (p_*)^{\gamma_0 \lambda}$. Due to the definition of the modified process D', if $Y_t = m$, then $Y_{t+1} = m$. Hence, the probability of reaching $Y_t = m$ is lower bounded by the probability of the event that the current level increases in all of at most m consecutive generations, i.e., $\Pr_t(Y_{t+m} = m) \geq (p_*)^{\gamma_0 \lambda m} > 0$. It follows that $\mathbb{E}[T] < \infty$.

By Theorem 26 and the upper bound on $g(0,1)$ in (15),

$$\begin{aligned}
\mathbb{E}[T] &\leq \lambda \cdot \frac{(1+\eta)g(0,1)}{\eta\varphi(1-\delta'')} \\
&< \frac{\lambda(1+\eta)}{(1-\delta'')} \cdot \frac{m2\eta\lambda^2 + \frac{4\varphi}{\lambda}\sum_{i=1}^{m-1}\frac{1}{z_i}}{\eta\varphi} \\
&= \frac{\lambda(1+\eta)}{(1-\delta'')} \cdot \left(\frac{2\lambda^2 m}{\varphi} + \frac{4}{\lambda\eta}\sum_{i=1}^{m-1}\frac{1}{z_i} \right) \\
&\leq \frac{\lambda}{(1-\delta'')} \left(\frac{2\lambda^2 m}{\delta(1-\delta')} + \frac{4}{1-(1+\delta)^{-1/2}} \sum_{i=1}^{m-1}\frac{1}{z_i} \right) \\
&\leq c''\lambda \left(\lambda^2 m + \sum_{i=1}^{m-1}\frac{1}{z_i} \right),
\end{aligned}$$

assuming that c'' is a sufficiently large constant. □

B Other proofs moved due to space limitations

Proof of Lemma 9. The conditions of Lemma 24 are satisfied. Hence, for $\delta := \delta_1\delta_2$, we get

$$\begin{aligned}
p_{\text{sel}}(R_0 \cup R_1) &= 1 - p_{\text{sel}}(R_2) \\
&\leq 1 - (1 + \delta_2(q_0 + q))p(R_2) \\
&= 1 - (1 + \delta_2(q_0 + q))(1 - q - q_0) \\
&= (q_0 + q)(1 - (1 - q - q_0)\delta_2) \\
&\leq (q_0 + q)(1 - \delta_1\delta_2) \\
&= p(R_0 \cup R_1)(1 - \delta).
\end{aligned}$$

□

Lemma 17 (Lemma 18 in Lehre (2011))**.** *If $Z \sim \text{Bin}(\lambda, r)$ with $r \geq \alpha(1+\delta)$, then for any $\kappa \in (0, \delta]$, $\mathbb{E}\left[e^{-\kappa X}\right] \leq e^{-\kappa\alpha\lambda}$.*

Lemma 18. *Consider any pair of independent binomial random variables* $X \sim \mathrm{Bin}(\lambda, p)$ *and* $Y \sim \mathrm{Bin}(\lambda, q)$, *where* $pq \geq (1+\sigma)^2 z$, $p, q, z \in (0,1)$ *and* $\sigma > 0$. *Then* $\mathbb{E}\left[e^{-\eta XY}\right] \leq e^{-\eta z \lambda^2}$ *for all* η *where* $0 < \eta \leq \frac{\sigma}{(1+\sigma)\lambda}$.

Proof. The proof applies Lemma 17 twice.

First, we apply Lemma 17 for the parameters $Z := X$, $\alpha := (z/q)(1+\sigma)$ and $\kappa := \eta Y$. The assumptions of the lemma then imply $p \geq \frac{z(1+\sigma)^2}{q} = \alpha(1+\sigma)$ and $\kappa \leq \frac{\sigma Y}{(1+\sigma)\lambda} \leq \sigma$, i.e., the conditions of Lemma 17 are satisfied. This then gives

$$\mathbb{E}\left[e^{-\eta XY} \mid Y\right] = \mathbb{E}\left[e^{-\kappa X} \mid Y\right] \leq e^{-\kappa\alpha\lambda} = \exp\left(-\frac{\eta z}{q}(1+\sigma)\lambda Y\right). \tag{23}$$

Secondly, we apply Lemma 17 for the parameters $Z := Y$, $\alpha := q/(1+\sigma)$ and $\kappa := \frac{z\eta}{q}(1+\sigma)\lambda$. We have $q = \alpha(1+\sigma)$, and by the assumption on η and the fact that $1 \geq q \geq z > 0$, it follows that

$$\kappa \leq \frac{\sigma}{(1+\sigma)\lambda}\frac{z}{q}(1+\sigma)\lambda \leq \sigma.$$

The conditions of Lemma 17 are satisfied, giving

$$\mathbb{E}\left[\exp\left(-\frac{\eta z}{q}(1+\sigma)\lambda Y\right)\right] = \mathbb{E}\left[e^{-\kappa Y}\right] \leq e^{-\kappa\alpha\lambda} \tag{24}$$

$$= \exp\left(-\frac{z\eta}{q}(1+\sigma)\lambda\frac{q}{1+\sigma}\lambda\right) = e^{-\eta z\lambda^2}. \tag{25}$$

By (23), (25), and the tower property of the expectation, it follows that

$$\mathbb{E}\left[e^{-\eta XY}\right] = \mathbb{E}\left[\mathbb{E}\left[e^{-\eta XY}\right] \mid Y\right] < e^{-\eta z\lambda^2}.$$

□

Lemma 19. *For any constant* $\delta_1 \in (0,1)$, *if* $1/3 < p_0 < 1-\delta_1$ *and* $q_0 \leq \delta_1/120$, *then there exists a constant* $\delta_1' > 0$ *such that*

$$\varphi := \frac{p_{sel}(S_0)}{p(S_0)}\frac{p_{sel}(R_1)}{p(R_1)} > 1 + \delta_1'.$$

Proof.

$$\begin{aligned}
\varphi &> \left(\frac{3}{2}(2-p_0)p_0(1-q) + q - 4q_0\right) \\
&\quad \times \frac{1}{2}\left((1-q_0)(3+q_0) - p_0(1-q_0(2+q_0))\right) \\
&> \frac{1}{4}\left(3(2-p_0)p_0(1-q) + 2q - 8q_0\right) \\
&\quad \times \left(3 - q_0(2+q_0) - p_0 + p_0q_0(2+q_0)\right) \\
&> \frac{1}{4}\left(3(2-p_0)p_0 + q(2-3(2-p_0)p_0) - 8q_0\right) \\
&\quad \times \left(3 - p_0 - 4q_0\right)
\end{aligned}$$

Considering the variable q independently, we distinguish between two cases.

Case 1: $2 < 3(2-p_0)p_0(1-q)$. In this case, the expression is minimised for $q = 1$, giving

$$\begin{aligned}\varphi &> \frac{1}{4}\left(2-8q_0\right)\left(3-p_0-4q_0\right)\\ &> \frac{1}{4}\left(2-8q_0\right)\left(2+\delta_1-4q_0\right)\\ &> \frac{1}{4}\left(2(2+\delta_1)-8q_0-(2+\delta_1)8q_0\right)\\ &> 1+\delta_1/2-8q_0\\ &> 1+\delta_1/4.\end{aligned}$$

Case 2: $2 \geq 3(2-p_0)p_0(1-q)$. In this case, the expression is minimised for $q = 0$, giving

$$\begin{aligned}\varphi &> \frac{1}{4}\left(3(2-p_0)p_0-8q_0\right)\left(3-p_0-4q_0\right)\\ &> \frac{3}{4}(2-p_0)p_0(3-p_0)-\frac{q_0}{4}\left(4\cdot 3(2-p_0)p_0+8(3-p_0)\right)\\ &> \frac{3}{4}(2-p_0)p_0(3-p_0)-12q_0\end{aligned}$$

Note that the function $f(x) := (2-x)(3-x)x$ has derivative $f'(x) < 0$ for $(5-\sqrt{7})/2 < x < 1$ and $f'(x) > 0$ if $1/3 < x < (5-\sqrt{7})/2$. Hence, to determine the minimum of the expression, it suffices to evaluate f at the extremal values $x = 1/3$ and $x = 1$, where $f(1/3) = 40/27$ and $f(1) = 2$. Hence, in case 2, we lower bound φ by $\varphi > \frac{3}{4}\cdot\frac{40}{27} - 12q_0 > 1 + \frac{1}{90}$. □

Lemma 20. *For any constant $\delta \in (0,1)$, if $p_0 q < \gamma_0 < 1-\delta$, $p_0 > 1-\delta/10$ and $q_0 < \delta/90$ then*

$$\frac{p_{sel}(S_0)}{p(S_0)}\frac{p_{sel}(R_1)}{p(R_1)} > 1 + \frac{23\delta}{300}.$$

Proof. Note first that the assumptions imply

$$3p_0q_0 < \delta/30. \tag{26}$$

When p_0 is sufficiently large, it suffices to only consider the cases where both x_1 and x_2 are selected in S_0. More precisely, conditional on the event $x_1 \in S_0 \wedge x_2 \in S_0$, the probability of selecting an element in R_1 is

$$\begin{aligned}&p_{\mathrm{sel}}(R_1 \mid x_1 \in S_0 \wedge x_2 \in S_0)\\ &\geq \Pr\left(y_1 \in R_1 \wedge y_2 \in R_1\right)\\ &\quad + \Pr\left(y_1 \in R_1 \wedge y_2 \in R_2\right)/2\\ &\quad + \Pr\left(y_1 \in R_2 \wedge y_2 \in R_1\right)\\ &= q^2 + q(1-q-q_0)/2 + (1-q-q_0)q\\ &= \frac{q}{2}(3-q-3q_0).\end{aligned}$$

Hence, the unconditional probability of selecting a pair in R_1 is

$$\begin{aligned}
p_{\text{sel}}(R_1) &> \frac{p_0^2 q}{2}\,(3 - q - 3q_0) \\
&> \frac{p_0 q}{2}\,(3(1-\delta/10) - (1-\delta) - 3p_0 q_0) \\
&> \frac{p_0 q}{2}\,(2 + \delta - \delta(3/10) - \delta/30) \\
&= p_0 q\,(1 + \delta/3)\,.
\end{aligned}$$

Using that $p(R_1) = q$, and $p_{\text{sel}}(S_0) \geq p_0^2$, we get

$$\begin{aligned}
\frac{p_{\text{sel}}(S_0)}{p(S_0)}\frac{p_{\text{sel}}(R_1)}{p(R_1)} &\geq \frac{p_0^2}{p_0}\frac{p_{\text{sel}}(R_1)}{p(R_1)} \\
&> (1-\delta/10)^2\,(1+\delta/3) \\
&= 1 + \frac{2\delta}{15} - \frac{17\delta^2}{300} + \frac{\delta^3}{300} \\
&> 1 + \frac{23\delta}{300}.
\end{aligned}$$

□

Lemma 21.

$$\varphi := \frac{p_{sel}(S_0)}{p(S_0)} \geq \frac{1}{2}\left((3+q_0)(1-q_0) - p_0(1 - q_0(2+q_0))\right)$$

Proof. Using Lemma 6, we get

$$\begin{aligned}
p_{\text{sel}}(S_0) = {} & \Pr(x_1 \in S_0 \wedge x_2 \in S_0) + \\
& + \Pr(x_1 \in S_0 \wedge x_2 \notin S_0) \\
& \quad \times \Pr((x_1, y_1) \succeq (x_2, y_2) \mid x_1 \in S_0 \wedge x_2 \notin S_0) \\
& + \Pr(x_1 \notin S_0 \wedge x_2 \in S_0) \\
& \quad \times (1 - \Pr((x_1, y_1) \succeq (x_2, y_2) \mid x_1 \notin S_0 \wedge x_2 \in S_0)) \\
\geq {} & \Pr(x_1 \in S_0 \wedge x_2 \in S_0) + \\
& + \Pr((x_1, y_1) \in S_0 \times R_1 \cup R_2 \wedge (x_1, y_1) \in S_1 \cup S_2 \times R_1 \cup R_2)/2 \\
& + \Pr(x_1 \notin S_0 \wedge x_2 \in S_0)(1 - \Pr(y_1 \in R_0 \wedge y_2 \in R_0)) \\
\geq {} & p_0^2 + p_0(1-p_0)(1-q_0)^2/2 + p_0(1-p_0)(1-q_0^2)
\end{aligned}$$

Recalling that $p(S_0) = p_0$, we get

$$\begin{aligned}
\varphi &\geq p_0 + (1-p_0)(1-q_0)^2/2 + (1-p_0)(1-q_0^2) \\
&= \frac{1}{2}\left((3+q_0)(1-q_0) - p_0(1-q_0(2+q_0))\right)
\end{aligned}$$

□

Lemma 22.

$$\varphi := \frac{p_{sel}(R_1)}{p(R_1)} > \frac{3}{2}(2 - p_0)p_0(1 - q) + q - 4q_0.$$

Proof. Using Lemma 6, we get

$$\begin{aligned}
p_{\mathrm{sel}}(R_1) = {} & \Pr(y_1 \in R_1 \wedge y_2 \in R_1) + \\
& + \Pr(y_1 \in R_1 \wedge y_2 \notin R_1) \\
& \quad \times \Pr((x_1, y_1) \succeq (x_2, y_2) \mid y_1 \in R_1 \wedge y_2 \notin R_1) \\
& + \Pr(y_1 \notin R_1 \wedge y_2 \in R_1) \\
& \quad \times (1 - \Pr((x_1, y_1) \succeq (x_2, y_2) \mid y_1 \notin R_1 \wedge y_2 \in R_1)) \\
\geq {} & \Pr(y_1 \in R_1 \wedge y_2 \in R_1) + \\
& + \Pr((x_1, y_1) \in S_0 \times R_1 \wedge (x_2, y_2) \in S_0 \times R_2)/2 \\
& + \Pr((x_1, y_1) \in S_0 \times R_1 \wedge (x_2, y_2) \in S_1 \cup S_2 \times R_2) \\
& + \Pr((x_1, y_1) \in S_1 \times R_1 \wedge (x_2, y_2) \in S_1 \times R_0)/2 \\
& + \Pr((x_1, y_1) \in S_1 \times R_1 \wedge (x_2, y_2) \in S_2 \times R_0) \\
& + \Pr((x_1, y_1) \in S_2 \times R_1 \wedge (x_2, y_2) \in S_2 \times R_0)/2 \\
& + \Pr(y_2 \in R_1) \\
& \quad \times (1 - \Pr(y_1 \in R_1) \\
& \qquad - \Pr((x_1, y_1) \in S_0 \times R_0 \wedge x_2 \in S_0) \\
& \qquad - \Pr((x_1, y_1) \in S_1 \times R_2 \wedge x_2 \in S_1 \cup S_2) \\
& \qquad - \Pr((x_1, y_1) \in S_2 \times R_2 \wedge x_2 \in S_2)) \\
\geq {} & q^2 + qp_0(1 - q - q_0)(p_0/2 + 1 - p_0) + \\
& + qpq_0(p/2 + 1 - p - p_0) + q(1 - p - p_0)^2 q_0/2 \\
& + q(1 - q - p_0^2 q_0 \\
& \quad - (1 - q - q_0)(p(1 - p_0) + (1 - p - p_0)^2)
\end{aligned}$$

Recalling that $p(R_1) = q$ and noting that $(4 - p_0)p_0 < 4$, it follows that

$$\begin{aligned}
\varphi > {} & \frac{3}{2}(2 - p_0)p_0(1 - q) + q \\
& + q_0 \left(\frac{3}{2} - (4 - p_0)p_0 \right) + p(1 - q - q_0)(1 - p - p_0) \\
> {} & \frac{3}{2}(2 - p_0)p_0(1 - q) + q - 4q_0.
\end{aligned}$$

□

Lemma 23. *If there exists a constant $\delta > 0$ such that*

1) $q_0 \leq \sqrt{2(1 - \delta)} - 1$

then

$$\varphi := \frac{p_{sel}(S_0 \cup S_1)}{p(S_0 \cup S_1)} > 1 + \delta(1 - p - p_0).$$

Proof. Using Lemma 6, we get

$$\begin{aligned}
&p_{\text{sel}}(S_0 \cup S_1) \\
&\quad = \Pr(x_1 \in S_0 \cup S_1 \wedge x_2 \in S_0 \cup S_1) + \\
&\qquad + \Pr(x_1 \in S_0 \cup S_1 \wedge x_2 \notin S_0 \cup S_1) \\
&\qquad\quad \times \Pr((x_1, y_1) \succeq (x_2, y_2) \mid x_1 \in S_0 \cup S_1 \wedge x_2 \notin S_0 \cup S_1) \\
&\qquad + \Pr(x_1 \notin S_0 \cup S_1 \wedge x_2 \in S_0 \cup S_1) \\
&\qquad\quad \times (1 - \Pr((x_1, y_1) \succeq (x_2, y_2) \mid x_1 \notin S_0 \cup S_1 \wedge x_2 \in S_0 \cup S_1)) \\
&\quad \geq \Pr(x_1 \in S_0 \cup S_1 \wedge x_2 \in S_0 \cup S_1) + \\
&\qquad + \Pr((x_1, y_1) \in S_0 \cup S_1 \times R_0 \cup R_1 \wedge (x_2, y_2) \in S_2 \times R_0 \cup R_1)/2 \\
&\qquad + \Pr(x_2 \in S_0 \cup S_1) \\
&\qquad\quad \times (1 - \Pr(y_2 \in R_0 \wedge (x_1, y_1) \in S_2 \times R_0) - \Pr(x_1 \in S_0 \cup S_1)) \\
&\quad \geq (p_0 + p)^2 + (p_0 + p)(1 - q_0)^2(1 - p - p_0)/2 + \\
&\qquad + (p_0 + p)(1 - (p_0 + p) - q_0^2(1 - p - p_0))
\end{aligned}$$

Recalling that $p(S_0 \cup S_1) = p_0 + p$, and the assumption of the lemma, it follows that

$$\begin{aligned}
\varphi &\geq 1 + (1 - p - p_0)((1 - q_0)^2/2 - q_0) \\
&= 1 + (1 - p - p_0)(1/2 - q_0(1 - q_0/2)) \\
&\geq 1 + (1 - p - p_0)(1/2 - (1 - 2\delta)/2) \\
&= 1 + \delta(1 - p - p_0).
\end{aligned}$$

□

Lemma 24. *If there exist a constant $\delta > 0$ such that*

1) $p_0 q = 0$.

2) $p_0 < \sqrt{2(1 - \delta)} - 1$

then

$$\varphi := \frac{p_{sel}(R_2)}{p(R_2)} \geq 1 + \delta(q_0 + q).$$

Proof. Using Lemma 6, we get

$$\begin{aligned}
&p_{\mathrm{sel}}(R_2)\\
&= \Pr\left(y_1 \in R_2 \wedge y_2 \in R_2\right) +\\
&\quad + \Pr\left(y_1 \in R_2 \wedge y_2 \notin R_2\right)\\
&\qquad \times \Pr\left((x_1, y_1) \succeq (x_2, y_2) \mid y_1 \in R_2 \wedge y_2 \notin R_2\right)\\
&\quad + \Pr\left(y_1 \notin R_2 \wedge y_2 \in R_2\right)\\
&\qquad \times (1 - \Pr\left((x_1, y_1) \succeq (x_2, y_2) \mid y_1 \notin R_2 \wedge y_2 \in R_2\right))\\
&\geq \Pr\left(y_1 \in R_2 \wedge y_2 \in R_2\right) +\\
&\quad + \Pr\left((x_1, y_1) \in S_1 \cup S_2 \times R_2 \wedge (x_2, y_2) \in S_1 \cup S_2 \times R_0 \cup R_1\right)/2\\
&\quad + \Pr\left(y_2 \in R_2\right)\\
&\qquad \times (1 - \Pr\left((x_1, y_1) \in S_0 \times R_1\right)\\
&\qquad\quad - \Pr\left((x_1, y_1) \in S_0 \times R_0 \wedge y_2 \in R_2\right) - \Pr\left(y_1 \in R_2\right))\\
&= (1 - q - q_0)^2\\
&\quad + (1 - q - q_0)(1 - p_0)^2(q_0 + q)/2\\
&\quad + (1 - q - q_0)(1 - p_0 q - p_0^2 q_0 - (1 - q - q_0))
\end{aligned}$$

From $p(R_2) = 1 - q - q_0$ and the assumptions of the lemma,

$$\begin{aligned}
\varphi &\geq 1 + (1 - p_0)^2(q_0 + q)/2 - p_0 q - p_0^2 q_0\\
&= 1 + (q_0 + q)/2 - p_0 q_0(1 + p_0/2)\\
&\geq 1 + (q_0 + q)/2 - q_0(1/2 - \delta)\\
&\geq 1 + (q_0 + q)\delta.
\end{aligned}$$

□

C Proof of the main theorem

Proof of Theorem 14. Note that we can guarantee $\chi \leq 1/(\alpha + \varepsilon)$, for a sufficiently small constant χ.

Define $\tau := c_0 r \lambda^3 n \in \mathrm{poly}(n)$ and $\tau_* := \min\{T/\lambda - 1, \tau\}$. We will condition on the event that $q_0 = 0$ holds for the first τ_* generations, and consider the run a failure otherwise. By Lemma 11, the probability of such a failure is no more than $\tau e^{-\Omega(\lambda)} + \tau e^{-\Omega(n)} = e^{-\Omega(\lambda)} + e^{-\Omega(n)}$, assuming that the constraint $\lambda \geq c \log(n)$ holds for a sufficiently large constant c.

We apply Theorem 3 with the $m = O(n)$ levels

$$(A_0^{(1)} \times B_0^{(1)}), \dots, (A_{(1-\beta)n}^{(1)}, B_{(1-\beta)n}^{(1)}),\\ (A_0^{(2)} \times B_0^{(2)}), \dots, (A_{(\alpha-\varepsilon)n}^{(2)}, B_{(\alpha-\varepsilon)n}^{(2)}),$$

defined in Section 5. It will suffice to choose the parameter γ_0 such that $1/3 < \gamma_0 \leq \sqrt{2(1-\delta)} - 1$ (e.g., $\gamma_0 = 2/5$ assuming that δ is sufficiently small).

We now prove conditions (G1), (G2a), and (G2b) separately for Phase 1 and Phase 2.

Phase 1: Assume that the current level belongs to phase 1 for any $j \in [0,(1-\beta)n]$. To prove that condition (G2a) holds, we will now show that the conditions of Lemma 12 are satisfied for the parameter $\psi := \gamma_0$. By Lemma 7, we have $p_0 < \gamma_0 \leq \sqrt{2(1-\delta)} - 1$, hence condition 1 is satisfied. Condition 2 is satisfied by the assumption on $q_0 = 0$. By the definition of the level, $(p_0 + p)(1 - q - q_0) < \gamma_0 = \psi$. Finally, we can assume that $p_0 q = 0$, otherwise the algorithm has already found an ε-approximate solution. All three conditions of Lemma 12 are satisfied. To produce an individual in $A^{(1)}_{j+1}$, it suffices to select and individual in $A^{(1)}_{j+1}$ and not mutate any of the bits, and analogously to produce an individual in $B^{(1)}_{j+1}$. In overall, for a sample $(x,y) \sim \mathcal{D}(P,Q)$, this gives

$$\begin{aligned}
&\Pr\left(x \in A^{(1)}_{j+1}\right) \Pr\left(y \in B^{(1)}_{j+1}\right) && (27)\\
&\geq p_{\mathrm{sel}}(A^{(1)}_{j+1}) p_{\mathrm{sel}}(B^{(1)}_{j+1}) \left(1 - \frac{\chi}{n}\right)^{2n} && (28)\\
&\geq (1 + \delta(1 - \sqrt{\gamma_0})) p(A^{(1)}_{j+1}) p(B^{(1)}_{j+1}) e^{-2\chi}(1 - o(1)) && (29)\\
&\geq (1 + \delta'')\gamma, && (30)
\end{aligned}$$

assuming sufficiently small mutation rate χ and parameter $\delta'' > 0$. Condition (G2a) of the level-based theorem is therefore satisfied for Phase 1. To prove (G2b), we apply Lemma 12 with parameter $\psi = \gamma_0(1+\delta)$. If $p(A^{(1)}_j)p(B^{(1)}_j) \geq \gamma_0(1+\delta)$, then (G2b) follows from the second statement of Lemma 12. If $\gamma_0 \leq p(A^{(1)}_j)p(B^{(1)}_j) \leq \gamma_0(1+\delta)$, then (G2b) follows from the first statement of Lemma 12.

Assume that $p(A^{(1)}_j \times B^{(1)}_j) = (p_0 + p)(1 - q - q_0) \geq \gamma_0$ and $(x,y) \sim \mathcal{D}(P,Q)$ Then, a P-individual can be obtained in $A^{(1)}_{j+1}$ by selecting an individual in $A^{(1)}_j$, and mutate one of $n - j \geq \beta n$ 1-bits, and no other bits, an event which occurs with probability at least

$$\begin{aligned}
\Pr\left(x \in A^{(1)}_{j+1}\right) &\geq p_{\mathrm{sel}}(A^{(1)}_j)(n-j)\frac{\chi}{n}\left(1 - \frac{\chi}{n}\right)^{n-1}\\
&\geq p_{\mathrm{sel}}(A^{(1)}_j) \cdot \Omega(1).
\end{aligned}$$

A Q-individual can be obtained in $B^{(1)}_{j+1}$ by selecting an individual in $B^{(1)}_{j+1}$ and not mutate any bits. By (30), this event occurs with probability at least

$$\Pr\left(y \in B^{(1)}_{j+1}\right) \geq p_{\mathrm{sel}}(B^{(1)}_j)\left(1 - \frac{\chi}{n}\right)^n \geq \frac{\Omega(1)}{p_{\mathrm{sel}}(A^{(1)}_j)}.$$

Hence, for a sample $(x,y) \sim \mathcal{D}(P,Q)$, we obtain

$$\Pr\left(x \in A^{(1)}_{j+1}\right) \Pr\left(y \in B^{(1)}_{j+1}\right) = \Omega(1),$$

hence condition (G1) will be satisfied for some parameter $z_j \in \Omega(1)$.

Phase 2: The analysis is analogous for this phase. To prove (G2a), assume that the current level belongs to phase 2 for any $j \in [0, (\alpha - \varepsilon)n]$. By the definitions of the levels in this phase, we must have $p_0 q(j-1) \geq \gamma_0$, thus $p_0 \geq \gamma_0 > 1/3$ where the last inequality follows from our choice of γ_0. Together with the assumption $q_0 = 0$, Corollary 13 gives

$$
\begin{aligned}
&\Pr\left(x \in A_j^{(2)}\right) \Pr\left(y \in B_j^{(2)}\right) && (31)\\
&\geq p_{\mathrm{sel}}(A_j^{(2)}) p_{\mathrm{sel}}(B_j^{(2)}) \left(1 - \frac{\chi}{n}\right)^{2n} && (32)\\
&\geq (1+\delta') p(A_j^{(1)}) p(B_j^{(1)}) e^{-2\chi}(1 - o(1)) && (33)\\
&\geq (1+\delta'')\gamma, && (34)
\end{aligned}
$$

assuming sufficiently small mutation rate χ and parameter $\delta'' > 0$. Condition (G2b) can be proved analogously to Phase 1, with the help of Corollary 13.

To prove condition (G1), we proceed as for Phase 1 and observe that to produce an individual in $A_{j+1}^{(2)}$, it suffices to select an P-individual in $A_j^{(2)}$ and not mutate any of the bits. To produce an individual in $B_{j+1}^{(2)}$, it suffices to select a Q-individual in $B_j^{(2)}$ and flip one of the at least $(1-\varepsilon)n = \Theta(n)$ number of 0-bits. Again, we obtain $\Pr\left(x \in A_{j+1}^{(2)}\right) \Pr\left(y \in B_{j+1}^{(2)}\right) = \Omega(1)$, hence condition (G1) will be satisfied for some parameter $z_j = \Omega(1)$.

Condition (G3) is satisfied as long as $\lambda \geq c' \log(m/z_*)$.

All the conditions are satisfied, and assuming that $q_0 = 0$, it follows that the expected time to reach an ε-approximation of Bilinear is no more than

$$
\mathbb{E}[T] \leq c''\lambda \left(\lambda^2 m + \sum_{i=1}^{m-1} \frac{1}{z_i}\right) = O(\lambda^3 n).
$$

By Markov's inequality, the probability that a solution has not been obtained in $O(r\lambda^3 n)$ time is less than $1/r$. Hence, in overall, we obtain for some constant $c_0 > 0$ and $\lambda \geq \log(n)$ and $\lambda \in \mathrm{poly}(n)$

$$
\Pr\left(T > c_0 r \lambda^3 n\right) \leq 1/r + e^{-\Omega(n)} + e^{-\Omega(\lambda)} \leq (1/r)(1 + o(1)).
$$

□

D Additional technical results

Lemma 25 (Dang and Lehre (2016)). *For $n \in \mathbb{N}$ and $x \geq 0$, we have* $1 - (1-x)^n \geq 1 - e^{-xn} \geq \frac{xn}{1+xn}$

Theorem 26 (Additive drift theorem Corus et al. (2018)). *Let $(Z_t)_{t\in\mathbb{N}}$ be a discrete-time stochastic process in $[0,\infty)$ adapted to any filtration $(\mathscr{F}_t)_{t\in\mathbb{N}}$. Define $T_a := \min\{t \in \mathbb{N} \mid Z_t \leq a\}$ for any $a \geq 0$. For some $\varepsilon > 0$ and constant $0 < b < \infty$, define the conditions*

1.1) $\mathbb{E}\left[Z_{t+1} - Z_t + \varepsilon\,; t < T_a \mid \mathscr{F}_t\right] \leq 0$ *for all* $t \in \mathbb{N}$*,*

1.2) $\mathbb{E}\left[Z_{t+1} - Z_t + \varepsilon\,; t < T_a \mid \mathscr{F}_t\right] \geq 0$ *for all* $t \in \mathbb{N}$*,*

2) $Z_t < b$ *for all* $t \in \mathbb{N}$*, and*

3) $\mathbb{E}\left[T_a\right] < \infty$*.*

If 1.1), 2), and 3) hold, then $\mathbb{E}\left[T_a \mid \mathscr{F}_0\right] \leq Z_0/\varepsilon$*.*
If 1.2), 2), and 3) hold, then $\mathbb{E}\left[T_a \mid \mathscr{F}_0\right] \geq (Z_0 - a)/\varepsilon$*.*

Lemma 27. *Let* X *and* Y *be two non-negative random variables with finite expectations. If* $X \succeq Y$*, then* $\mathbb{E}\left[X\right] \geq \mathbb{E}\left[Y\right]$*.*

Proof. By definition, $X \succeq Y$ implies $\Pr\left(Y \leq z\right) \geq \Pr\left(X \leq z\right)$ for all $z \in \mathbb{R}$. Using that X and Y are non-negative random variables,

$$\mathbb{E}\left[X\right] = \int_0^\infty 1 - \Pr\left(X \leq z\right) dz \geq \int_0^\infty 1 - \Pr\left(Y \leq z\right) dz = \mathbb{E}\left[Y\right].$$

□