
Theoretical Guarantees for the Retention of Strict Nash Equilibria by Coevolutionary Algorithms

Alistair Benford

School of Computer Science
University of Birmingham
a.s.benford@bham.ac.uk

Per Kristian Lehre

School of Computer Science
University of Birmingham
p.k.lehre@bham.ac.uk

Abstract

Most methods for finding a Nash equilibrium rely on procedures that operate over the entire action space, making them infeasible for settings with too many actions to be searched exhaustively. Randomised search heuristics such as coevolutionary algorithms offer benefits in such settings, however they lack many of the theoretical guarantees established for exhaustive methods such as zero-regret learning. We address this by developing a method for proving necessary and sufficient conditions for a coevolutionary algorithm to be stable, in the sense that it reliably retains a Nash equilibrium following discovery. As the method provides bounds that are adapted to both application and algorithm instance, it can be used as a practical tool for parameter configuration. We additionally show how bounds on regret may be deduced from our results and undertake corresponding empirical analysis.

1 Introduction

As many challenging problems are characterised by the presence of competition between agents, the demand for efficient techniques for the determination of Nash equilibria has brought about a range of powerful machine learning techniques, including Q-Learning [10, 20, 33], Fictitious Self-Play [7, 25], Counterfactual Regret Minimisation [8, 9, 64], and gradient-based approaches [4, 19, 57, 63]. Analysis of regret for algorithms such as EXP3 [3] and FTRL [23] provide theoretical guarantees of convergence to a Nash equilibrium in the zero-sum adversarial multi-armed bandit setting, however such algorithms rely on procedures operating over the entire strategy space (such as storing a vector over all available actions). In many settings, such as those characterised by a sequence of binary decisions, a combinatorial explosion in the number of actions quickly make any exhaustive approach infeasible. These many-action settings can only be addressed using procedures that exploit underlying topological features of the strategic representation and corresponding payoff landscape in order to guide a search towards a Nash equilibrium. Coevolutionary algorithms (CoEAs), a form of evolutionary algorithms (EAs) employing two or more competing populations, are one such approach, and recent theoretical work proves that CoEAs can efficiently discover the Nash equilibrium of a range of two-player many-action games defined over bitstrings [5, 30, 31, 32, 40, 43].

The successful application of adversarial and self-play techniques is uniquely challenging due to the potential for pathologies such as cycling or instability, and such pathologies have been observed in a range of such methods including GANs [53, 62], AlphaStar [60], SGA [4], as well as CoEAs in general [18]. In the case of CoEAs, one major open problem is the question of when to terminate and extract an output from the current populations. This is an important consideration even for standard evolutionary algorithms in non-adversarial settings [46], and made even more challenging for CoEAs as the absence of a unary fitness function makes the identification of best-so-far strategies both expensive and subjective. Thus, to maintain confidence in the output of a CoEA, one should ensure that the algorithm is designed and configured in such a way that any discovered Nash equilibrium will

not be forgotten following discovery. Algorithms with this property are more amenable to maintain progress towards the Nash equilibrium even before discovery (as per conditions of the level-based theorem for CoEAs [40]), making its analysis even more essential.

The question of how CoEAs reach a Nash equilibrium is addressed theoretically by *runtime analysis*, which is the subject of all existing theory for CoEAs. For an algorithm to have low runtime, it is often necessary for the strength of its mutation operator (i.e., its *mutation rate*) to be sufficiently low relative to selective pressure (although this alone is typically not sufficient) [38, 40, 50]. However, there is a gap in our understanding of algorithm behaviour following the time after which a Nash equilibrium first enters the populations. This paper presents a theoretical basis for the determination of conditions for a CoEA to be ‘stable’ in the sense that it is overwhelmingly likely to retain a Nash equilibrium following discovery. In particular, we quantify the link between the stability of a CoEA and the mutation rate, and this link will closely depend on the selection process being used (similar to runtime results that also impose necessary conditions on mutation rate). We will later see (Theorem 3.5) that a given algorithm will never be stable if mutation is too strong. As a very low mutation rate can inhibit exploration, knowledge about how strong mutation can be while still ensuring stability is extremely valuable for practitioners. Our contributions are thus as follows.

1. An easily applicable theoretical technique (Lemmas 3.3 and 3.4) for ascertaining upper and lower bounds on the threshold between stability and instability for population-based CoEAs with unary variation.
2. Derivation of stability bounds using this technique for large classes of CoEAs (Theorems 3.5 and 3.6) as well as two specific CoEAs (Theorems 3.7 and 3.8). We also confirm the predictions made by these derived bounds empirically.
3. A proof that stability guarantees can be combined with runtime analysis to additionally quantify the regret of CoEAs, akin to the regret guarantees established for learning algorithms on small action spaces.

1.1 Related work

CoEAs have been used in numerous applications to adversarial optimisation and multi-agent learning, including generative adversarial networks [14], cybersecurity and defence [21, 26, 29, 59, 65], and detecting tax evasion [27]. Coevolution has extensive use for the training of game-playing agents in studies on backgammon [54], poker [49], Atari games [36], and StarCraft II [2, 60], to name just a few. Other applications of coevolution introduce competition to improve learning by coevolving a population of environments alongside agents [1, 12, 61]. For a general overview of fundamentals of CoEAs, their use cases, and challenges arising in their application, we refer the reader to [55].

CoEAs can be viewed as an extension of EAs, for which there has been a large amount of theoretical analysis (see survey [17] for example). Despite demand (see [55]), similar analysis for CoEAs is more limited. Nonetheless, a rigorous theory for competitive CoEAs was initiated by Lehre [39, 40] who provided guarantees for the expected time for an algorithm called PDCoEA (which we consider later in Section 3.2) to discover the Nash equilibrium of a bilinear game (which we consider in our empirical analysis in Section 4). Subsequent theoretical work analyses the impact of algorithm design on runtime [30, 31, 32] and the use of CoEAs for the discovery of optimal play for new games and problem classes, including symmetric zero-sum games [5], combinatorial games [6], binary test-based problems [42, 43], and potential games [28].

Some theoretical results for EAs establish an ‘error threshold’ expressed as a function of selective pressure. Configuring mutation rate above an error threshold makes it difficult to sustain a positive proportion of individuals inside a region of the search space containing the target. In previous analysis this is significant because it can lead to exponential runtime, however it is directly related to stability results here which establish thresholds on mutation rate, also expressed in terms of selection, above which a CoEA struggles to maintain populations on the solution concept. Error thresholds are standard in analysis of EAs (see, for example, [38, 50]), and have also been proven to exist for CoEAs [40]. The role that error thresholds play in coevolution has also been analysed empirically in [29] where it was observed that the mutation rate of self-adaptive version of PDCoEA converged to a value just below the error threshold without previous knowledge.

1.2 Notation

A *probability distribution* over a finite set S is a function $p : S \rightarrow [0, 1]$ which satisfies $\sum_{s \in S} p(s) = 1$, and $\mathcal{P}(S)$ will be used to denote the set of probability distributions over S . An S -valued random variable x is *distributed according to* p (written $x \sim p$), if $\mathbb{P}(x = s) = p(s)$ holds for every $s \in S$. We use $x = \operatorname{argmax}_{y \in S} f(y)$ as a shorthand for saying that x is sampled uniformly from the set $\{y \in S : f(y) = \max_{z \in S} f(z)\}$. Given a set \mathcal{X} and a natural number $\lambda \in \mathbb{N}$, we will use \mathcal{X}^λ to denote the set of tuples of length λ over \mathcal{X} . Given a tuple $P := (P(1), \dots, P(\lambda)) \in \mathcal{X}^\lambda$, we often regard P also as a multiset (that is a subset of \mathcal{X}) of cardinality λ , and hence write statements such as $x \in P$ to mean $x \in \{P(i) : i \in [\lambda]\}$, or use $|P \cap A|$ to represent the quantity $|\{i \in [\lambda] : P(i) \in A\}|$.

2 Setting

The setting we consider is a conventional two-player game scenario, however it readily applies not only to game-playing applications, but also to real-world adversarial optimisation, scenario optimisation, multi-agent systems, and settings with competition added to enhance learning.

Definition 2.1. A problem (or game) is a tuple $(\mathcal{X}, \mathcal{Y}, f_1, f_2, A \times B)$ where $f_1 : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, $f_2 : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, $A \subseteq \mathcal{X}$, and $B \subseteq \mathcal{Y}$. f_i is called the payoff function for player i and $A \times B$ is called the solution concept. A problem class is simply a collection of problems.

Let \mathcal{G} represent the class of all problems. As Definition 2.1 imposes no conditions or restrictions on the sets A and B , there is no assurance that a problem (in the strict sense of the definition) has an associated solution concept that is of meaningful or practical. In reality, we will restrict attention to classes of problems which have solution concepts that fit a common understanding of what it means to be strategically optimal. In particular, we will use STRICTNASH to denote the class of problems for which $f_1(x, y) > f_1(x', y)$ and $f_2(x, y) > f_2(x, y')$ holds for all $x \in A$, $x' \notin A$, $y \in B$, $y' \notin B$, and focus on this problem class as a particular case of interest. Nonetheless, the high degree of generality in Definition 2.1 allows for results that potentially apply to alternative formulations of optimality. For example, some of our negative results are general enough to apply to *any* solution concept that is a singleton set. Therefore, let SINGLETONSOLUTION be the class of problems with $|A| = |B| = 1$. It is also common to prove results for classes of problems with additional structure imposed on the payoff function or strategy spaces. Let ZEROSUM be the class of problems for which $f_1(x, y) + f_2(x, y) = 0$ holds for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Given $n \in \mathbb{N}$, let BITSTRING $_n$ be the class of problems with $\mathcal{X} = \mathcal{Y} = \{0, 1\}^n$.

We consider CoEAs consisting of two populations $P \subseteq \mathcal{X}$ and $Q \subseteq \mathcal{Y}$ (informally labelled the *predator* and *prey* populations) constructed using two fundamental randomised procedures: *selection* and *mutation*. Selection uses evaluations of the payoff function to determine which individuals are best suited to be parents for the next generation. Mutation randomly varies the genotype of those selected parents to produce offspring. Here we formalise these two components and characterise some key properties before stating the algorithmic model to which our results apply.

Definition 2.2. Given a search domain \mathcal{X} , a mutation operator (or unary variation operator) over \mathcal{X} is a function $\mathcal{M}_\mathcal{X} : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{X})$. Given search domains \mathcal{X} and \mathcal{Y} and a subset $R \subseteq [0, 1]$, we use $\text{MUTATION}(\mathcal{X}, \mathcal{Y}; R)$ to denote the set of pairs $(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y})$ of mutation operators over \mathcal{X} and \mathcal{Y} that satisfy the following statisticity condition.

$$\mathcal{M}_\mathcal{X}(x)(x) \cdot \mathcal{M}_\mathcal{Y}(y)(y) \in R \quad \text{for all } (x, y) \in \mathcal{X} \times \mathcal{Y}. \quad (1)$$

In our results, the set R will characterise the strength of the mutation operators being discussed. For example, if $(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (q_0, 1])$, then (1) ensures the probability that mutation has no effect on a predator-prey pair greater than q_0 ; in this context, q_0 represents a bound on the strength of the mutation operators being used. For blackbox optimisation problems over bitstrings, it is standard to use mutation operators which are *unbiased*, in the sense that the effect of mutation is preserved under permutations of the bit positions and applications of binary masks (see [44] for a precise definition). Going further, with no a priori information about how the payoff landscape differs between players, it is natural when $\mathcal{X} = \mathcal{Y}$ to choose $\mathcal{M}_\mathcal{X} = \mathcal{M}_\mathcal{Y}$. Thus we will often consider the following subclass of $\text{MUTATION}(\{0, 1\}^n, \{0, 1\}^n; R)$ (where the final assumption in Definition 2.3 simply ensures the probability of mutating to the most distant searchpoint is small, as is standard).

Definition 2.3. Given $n \in \mathbb{N}$ and $R \subseteq [0, 1]$, we use $\text{UNBIASED}(n; R)$ to denote the set of $(\mathcal{M}_{\mathcal{X}}, \mathcal{M}_{\mathcal{Y}}) \in \text{MUTATION}(\{0, 1\}^n, \{0, 1\}^n; R)$ for which $\mathcal{M}_{\mathcal{X}}$ is unbiased, $\mathcal{M}_{\mathcal{Y}} = \mathcal{M}_{\mathcal{X}}$, and $\mathcal{M}_{\mathcal{X}}(\mathbf{0}^n)(\mathbf{1}^n) \leq 1/n$.

Given populations $(P, Q) \in \mathcal{X}^\lambda \times \mathcal{Y}^\lambda$, the role of a selection operator is to stochastically select individuals (here, one pair at a time) from the populations that are suitable parents for the following generation. The defining property of a selection operator is that its output must be an exact copy of a pair from the input populations [52], defined formally as follows.

Definition 2.4. A selection operator is a function $\mathcal{S}_\lambda : \mathcal{X}^\lambda \times \mathcal{Y}^\lambda \rightarrow \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ (defined for each $\lambda \in \mathbb{N}$) such that for every $(P, Q) \in \mathcal{X}^\lambda \times \mathcal{Y}^\lambda$, if $(x, y) \sim \mathcal{S}_\lambda(P, Q)$ then $\mathbb{P}(x \in P \wedge y \in Q) = 1$.

Selection operators use the payoff functions f_1 and f_2 (as well as the search domains \mathcal{X} and \mathcal{Y}) to make decisions about what to select. Thus when we say ‘selection operator’, we typically mean ‘a description of a selection operator for every possible problem instance $G \in \mathcal{G}$ ’. Algorithm 1 describes a model for non-elitist population-based CoEAs with unary variation. We note that this model is a coevolutionary analogue to the Population Selection-Variation Algorithm (PVSA) for standard EAs [38, 52], and additionally fits the population-based coevolutionary process model studied in [40].

Algorithm 1 Population-based CoEA with unary variation

Require: Population size λ
Require: Initial populations $P_0 \in \mathcal{X}^\lambda$ and $Q_0 \in \mathcal{Y}^\lambda$
Require: Selection operator \mathcal{S}_λ
Require: Mutation operators $\mathcal{M}_{\mathcal{X}} : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{X})$ and $\mathcal{M}_{\mathcal{Y}} : \mathcal{Y} \rightarrow \mathcal{P}(\mathcal{Y})$

- 1: **for** $t \in \mathbb{N}$ until termination criterion met **do**
- 2: **for** $j \in [\lambda]$ **do**
- 3: Sample $(x, y) \sim \mathcal{S}_\lambda(P_t, Q_t)$
- 4: Sample $P_{t+1}(j) \sim \mathcal{M}_{\mathcal{X}}(x)$
- 5: Sample $Q_{t+1}(j) \sim \mathcal{M}_{\mathcal{Y}}(y)$
- 6: **end for**
- 7: **end for**

Typically, search domains and mutation operators are application-specific, and population size λ is considered to be an algorithmic parameter. Thus, we will generally consider CoEAs to be defined solely by their selection operator \mathcal{S}_λ . In light of this, we will use $\mathcal{A}_S(\mathcal{M}_{\mathcal{X}}, \mathcal{M}_{\mathcal{Y}}, \lambda)$ to denote the population-based CoEA with unary variation corresponding to selection operator \mathcal{S}_λ (where we drop the implicit λ for brevity), mutation operators $\mathcal{M}_{\mathcal{X}}$ and $\mathcal{M}_{\mathcal{Y}}$, and population size λ .

3 Stability analysis

Roughly, we will say a CoEA is γ -stable on a problem if given it is initialised with at least γ proportion of $P_0 \times Q_0$ on the solution concept, with overwhelmingly high probability it will continue to maintain a γ proportion of $P_t \times Q_t$ on the Nash for an overwhelmingly long period of time (where ‘overwhelmingly’ here denotes exponential with respect to population size). This notion is formalised using the following definitions.

Definition 3.1. Given an algorithm \mathcal{A} running on a problem G with solution concept S , we define the γ -departure and γ -hitting times to be

$$T_{\text{dep}}^\gamma(\mathcal{A}; G) = \min \{t \geq 1 : |(P_t \times Q_t) \cap S| \leq \gamma\lambda^2\},$$

$$T_{\text{hit}}^\gamma(\mathcal{A}; G) = \min \{t \geq 1 : |(P_t \times Q_t) \cap S| \geq \gamma\lambda^2\}.$$

That is, $T_{\text{dep}}^\gamma(\mathcal{A}; G)$ is the first time (excluding $t = 0$) for which the populations of \mathcal{A} have at most than a γ proportion of pairs on the solution concept for G , and $T_{\text{hit}}^\gamma(\mathcal{A}; G)$ is the first time (excluding $t = 0$) for which the algorithm’s populations have at least a γ proportion of pairs on the solution concept for G .

Definition 3.2. A population-based CoEA \mathcal{A} with population size λ is γ -stable on a problem G with solution concept S if there exists $\delta > 0$ which does not depend on λ such that, for all populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ satisfying $|(P \times Q) \cap S| \geq \gamma\lambda^2$, it holds that

$$\mathbb{P}[T_{\text{dep}}^\gamma(\mathcal{A}; G) \leq e^{\delta\lambda} \mid (P_0, Q_0) = (P, Q)] \leq e^{-\delta\lambda}. \quad (2)$$

Definition 3.2 is the strongest that can reasonably be adopted, in the sense if $e^{\delta\lambda}$ were replaced by any function $h : \mathbb{N} \rightarrow \mathbb{R}$ that grows faster than exponential with respect to λ , then it would be impossible for any non-trivial population-based CoEA with unary variation to satisfy the definition. This observation is proven formally in Appendix A.

3.1 General tools

We now present the main tools used to derive stability bounds for population-based CoEAs. The first of these is a lemma that produces a bound $q_0(\gamma)$ on the strength of the mutation operators that can be used while still ensuring γ -stability. The resulting bound is dependent on the description of the algorithm's selection operator \mathcal{S}_λ , or more precisely it is dependent on two functions $\alpha : [0, 1]^2 \rightarrow [0, 1]$ and $\beta : [0, 1]^2 \rightarrow [0, 1]$ that can be straightforwardly derived from a description of the selection operator. Semantically, $\alpha(a, b)$ represents a lower bound the probability that a selected predator lies on the Nash equilibrium, given an a -proportion of predators and b -proportion of preys in the present populations lie on the Nash (and similar for $\beta(a, b)$ and a selected prey). Later, we will see that the application-specific expressions for α and β are typically polynomials and that deriving them is a matter of routine computation. Obtaining expressions for α and β is one of only two application-specific steps required to apply this tool. The other is to compute the maximisation set out in (3); however, as α and β are typically polynomials (and also often exhibit symmetries that simplify analysis) this is most often a case of applying elementary calculus and otherwise relying on numerical solvers. The proof of Lemma 3.3 is provided alongside a more general version of the tool in Appendix B.3.

Lemma 3.3. *Let \mathcal{S}_λ be a selection operator, let $G := (\mathcal{X}, \mathcal{Y}, f_1, f_2, A \times B)$, and let $\gamma \in [0, 1]$. Suppose there exist increasing functions $\alpha : [0, 1]^2 \rightarrow [0, 1]$ and $\beta : [0, 1]^2 \rightarrow [0, 1]$ such that, for all $\lambda \in \mathbb{N}$ and populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$,*

$$\mathbb{P}(x \in A) \geq \alpha\left(\frac{|P \cap A|}{\lambda}, \frac{|Q \cap B|}{\lambda}\right) \quad \text{and} \quad \mathbb{P}(y \in B) \geq \beta\left(\frac{|P \cap A|}{\lambda}, \frac{|Q \cap B|}{\lambda}\right)$$

holds for $(x, y) \sim \mathcal{S}_\lambda(P, Q)$. Then, provided $(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (q_0(\gamma), 1])$ where

$$q_0(\gamma) := \sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)}, \quad (3)$$

$\mathcal{A}_\mathcal{S}(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}, \lambda)$ is γ -stable on G .

We note that the assumptions in Lemma 3.3 imply condition (G2b) in the level-based theorem [40, Theorem 3] used to derive bounds on the expected runtime of CoEAs. For details, see Appendix B.3.

As well as characterising regimes where stability is guaranteed, we must also assess when stability is not guaranteed (or more drastically, when instability is guaranteed). This is possible using the second of our main tools, Lemma 3.4. Its first conclusion (A1) is a direct analogue to Lemma 3.3, and the outstanding application-specific steps are also the same (except that α and β must be obtained as upper bounds on selection probabilities instead of lower). However, Lemma 3.4 is extended in two further ways. First, it is stated with respect to bitstring problems and unbiased mutation operators. As the purpose of the tool is to identify unstable circumstances, this is important that such circumstances are natural applications rather than unrealistic instances without wider significance, and a restriction to unbiased mutation over bitstrings is one well studied way to ensure this. Nonetheless, in order to have negative results with utility beyond bitstring games, we state a more general version of the result applicable to arbitrary domains in Appendix B.4 and use it to derive Lemma 3.4. Second, the additional conclusion A2 identifies circumstances where an algorithm is unstable in a very strong and critical sense in that the proportion of its populations on the Nash drops to almost zero in a small amount of time. The proof of Lemma 3.4 is found alongside its generalisation in Appendix B.4.

Lemma 3.4. *Let $q, \gamma \in [0, 1]$ and \mathcal{S}_λ be a selection operator. Let $(\mathcal{G}_n)_{n=1}^\infty$ be a sequence of problem classes with $\mathcal{G}_n \subseteq \text{BITSTRING}_n \cap \text{SINGLETONSOLUTION}$. Suppose $K \in \mathbb{R} \cup \{\infty\}$ is a constant and $\alpha : [0, 1]^2 \rightarrow [0, 1]$ and $\beta : [0, 1]^2 \rightarrow [0, 1]$ are continuous increasing functions such that, for all $\lambda, n \in \mathbb{N}$, problems $(\mathcal{X}, \mathcal{Y}, f_1, f_2, (x^*, y^*)) \in \mathcal{G}_n$, and populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$,*

$$\mathbb{P}(x = x^*) \leq \alpha\left(\frac{|P \cap \{x^*\}|}{\lambda}, \frac{|Q \cap \{y^*\}|}{\lambda}\right) \quad \mathbb{P}(y = y^*) \leq \beta\left(\frac{|P \cap \{x^*\}|}{\lambda}, \frac{|Q \cap \{y^*\}|}{\lambda}\right)$$

$$\mathbb{P}((x, y) = (x^*, y^*)) \leq K \cdot \frac{|(P \times Q) \cap \{(x^*, y^*)\}|}{\lambda^2}$$

holds for $(x, y) \sim \mathcal{S}_\lambda(P, Q)$. Then the following results hold.

A1 *Provided*

$$q < \sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)}, \quad (4)$$

there is a constant $n_0 \in \mathbb{N}$ such that for any $n \geq n_0$, $G \in \mathcal{G}_n$, $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{UNBIASED}(n; [0, q])$, the algorithm $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is not γ -stable on any $G \in \mathcal{G}_n$.

A2 *Provided* $K < \infty$, $\alpha(a, b) = \beta(b, a)$ for all $a, b \in [0, 1]$, and that

$$q < \inf_{(a, b) \in (0, \sqrt{\gamma}]^2} \frac{ab}{\alpha(a, b) \cdot \alpha(b, a)}, \quad (5)$$

there is a constant $\delta > 0$ such that the following holds for all functions $\tau : \mathbb{N} \rightarrow \mathbb{R}$ and $\varepsilon : \mathbb{N} \rightarrow \mathbb{R}$ with $\varepsilon(n) \geq \frac{10}{\delta n}$. For any $n \geq 2/(\delta\sqrt{\gamma})$, $G \in \mathcal{G}_n$, $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{UNBIASED}(n; [0, q])$, λ satisfying

$$\lambda \geq \max \left\{ \frac{K}{\delta}, \frac{2}{\delta^2 \gamma} \log(\delta \tau(n)^2 \varepsilon(n)) \right\}, \quad (6)$$

and populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ with $|P \cap \{x^*\}|, |Q \cap \{y^*\}| \leq \sqrt{\gamma} \lambda$, it holds that

$$\mathbb{P}[T_{\text{dep}}^{\varepsilon(n)}(\mathcal{A}; G) \geq \tau(n) \mid (P_0, Q_0) = (P, Q)] \leq \frac{3}{\delta \varepsilon(n) \tau(n)},$$

where $\mathcal{A} := \mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$. In particular, choosing ε and τ appropriately (e.g., $\varepsilon(n) = \frac{6}{\delta n}$ and $\tau(n) = n^2$) shows that \mathcal{A} is not γ -stable on any $G \in \mathcal{G}_n$ for n sufficiently large.

3.2 Applications

Here we showcase the utility of the tools from Section 3.1 by applying them to derive stability bounds for a range of algorithms and problem classes. First, we show that it is immediate from Lemma 3.4 that γ -stability is never possible if $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{UNBIASED}(n; [0, \gamma])$.

Theorem 3.5. *Let S_λ be any selection operator. If $\gamma \in [0, 1]$ and $q < \gamma$, then the conclusion of **A1** holds for $\mathcal{G}_n = \text{BITSTRING}_n \cap \text{SINGLETONSOLUTION}$.*

Proof. The prerequisites of Lemma 3.4 are satisfied with $\alpha(a, b) = \beta(a, b) = 1$ and $K = \infty$. \square

Improvements over this universal bound arise from the introduction of an additional weak assumption on the structure of the selection operator. A feature common to many selection operators is to first sample several candidates from each population uniformly at random (with replacement), and then make an informed selection from those candidates. We will say that a k -candidate selection operator is one that makes the predator selection from k predator candidates and the prey selection from k prey candidates. This informal description is sufficient for the sake of the material in this paper; however, for the sake of rigour a formal description may be found in Appendix B.5.

Theorem 3.6. *Let S_λ be a k -candidate selection operator.*

B1 *If $\gamma \in [0, 1]$ and $q < \gamma/(1 - (1 - \gamma)^k)$, then the conclusion of **A1** holds for $\mathcal{G}_n = \text{BITSTRING}_n \cap \text{SINGLETONSOLUTION}$.*

B2 *If $\gamma \in [0, 1]$ and $q < k^{-2}$, then the conclusion of **A2** holds for $\mathcal{G}_n = \text{BITSTRING}_n \cap \text{SINGLETONSOLUTION}$.*

Proof. k -candidate selection operators satisfy the prerequisites of Lemma 3.4 with $\alpha(a, b) = 1 - (1 - a)^k$, $\beta(a, b) = 1 - (1 - b)^k$, and $K = k^2$; see Appendix B.5 for further details. \square

Theorems 3.5 and 3.6 characterise regimes where instability is inevitable (and hence should be avoided when parameterising algorithms) for highly general classes of selection operators. By considering specific selection operators, it is also possible to apply our main tools to derive not only stronger bounds on instability, but also practical guidelines of how to parameterise mutation to assure stability. We carry out these derivations for two specific population-based CoEAs: Pairwise

Dominance CoEA (PDCoEA) and Tournament Selection CoEA (TSCoEA)¹. The selection operators defining each algorithm are given below.

Algorithm 2 Selection operator for PDCoEA

```

1: Sample  $x_1, x_2 \sim \text{Unif}(P)$ 
2: Sample  $y_1, y_2 \sim \text{Unif}(Q)$ 
3: if  $f_1(x_1, y_1) \geq f_1(x_2, y_1)$  and  $f_2(x_1, y_1) \geq f_2(x_2, y_2)$  then
4:   Set  $(x, y) = (x_1, y_1)$ 
5: else
6:   Set  $(x, y) = (x_2, y_2)$ 
7: end if

```

Algorithm 3 Selection operator for TSCoEA

```

1: Sample  $x_1, \dots, x_k \sim \text{Unif}(P_t)$ 
2: Sample  $y'_1, \dots, y'_\ell \sim \text{Unif}(Q_t)$ 
3: Set  $i = \text{argmax}_{r \in [k]} \min_{s \in [\ell]} f_1(x_r, y'_s)$ 
4: Sample  $y_1, \dots, y_k \sim \text{Unif}(Q_t)$ 
5: Sample  $x'_1, \dots, x'_\ell \sim \text{Unif}(P_t)$ 
6: Set  $j = \text{argmax}_{r \in [k]} \min_{s \in [\ell]} f_2(x'_s, y_r)$ 
7: Set  $(x, y) = (x_i, y_j)$ 

```

In each selection, PDCoEA decides between two sampled pairs (x_1, y_1) using a pairwise dominance relation. PDCoEA has been studied empirically on game theoretic attacker-defender cyber security interactions [29, 41], and theoretically on pseudoboolean payoff functions [40, 43]. In Theorem 3.7 we apply Lemma 3.3 to derive conditions for the assured stability of PDCoEA for three increasingly specialised problem classes: STRICTNASH, STRICTNASH \cap SINGLETONSOLUTION, and STRICTNASH \cap SINGLETONSOLUTION \cap ZEROSUM. Going further, in the case of STRICTNASH \cap SINGLETONSOLUTION, we show that the proven bound is best possible using both conclusions of Lemma 3.4. Illustrations of the trade-off between mutation strength and γ -stability for the latter two classes are given in Figure 1 and a proof of the theorem can be found in Appendix B.6.

Theorem 3.7. *Let S_λ be the selection operator for PDCoEA (see Algorithm 2) and let $\gamma \in (0, 1]$.*

C1 *Let $q_0(\gamma) = (\frac{5}{2}\sqrt{\gamma} - \frac{3}{2}\gamma)^{-2}$. If $G \in \text{STRICTNASH}$ and $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (q_0(\gamma), 1])$ then $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is γ -stable on G .*

C2 *Let $q_0(\gamma) = (3\sqrt{\gamma} - 2\gamma)^{-2}$.*

C2.1 *If $G \in \text{STRICTNASH} \cap \text{SINGLETONSOLUTION}$ and $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (q_0(\gamma), 1])$ then $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is γ -stable on G .*

C2.2 *There is a sequence of problems $(G_n)_{n=1}^\infty$ with $G_n \in \text{STRICTNASH} \cap \text{SINGLETONSOLUTION} \cap \text{BITSTRING}_n$ for every $n \in \mathbb{N}$ such that if $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{UNBIASED}(n; [0, q_0(\gamma)))$, then $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is not γ -stable on G_n for n sufficiently large.*

C2.3 *If $\gamma \in (0, 9/16]$ and $q \in [0, q_0(\gamma))$, then the conclusion of **A2** holds for $\mathcal{G}_n = \{G_n\}$ (where $(G_n)_{n=1}^\infty$ is the problem sequence arising in **C2.2**).*

C3 *Let $\alpha_\sigma(a, b) = a(a + (1 - a)(3b - b^2 + (1 - b)^2\sigma(1 + \frac{1}{2}\sigma)))$ and*

$$q_0(\gamma) = \sup_{a \in [\gamma, 1]} \sup_{\sigma \in [0, 1]} \frac{\gamma}{\alpha_\sigma(a, \gamma/a) \cdot \alpha_{1-\sigma}(\gamma/a, a)}.$$

If $G \in \text{STRICTNASH} \cap \text{SINGLETONSOLUTION} \cap \text{ZEROSUM}$ and $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (q_0(\gamma), 1])$ then $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is γ -stable on G .

TSCoEA [31] is archetypical of a broad class of CoEAs which perform a k -tournament selection using some chosen metric calculated based on payoffs against a number of opponents (see [22, 35, 58] for examples of similar such algorithms). While any desired metric may be chosen (such as best average payoff or membership of the mixed Nash equilibrium of the implied restricted game), we focus on best worst-case (or max-min) payoff for two good reasons. First, it can be computed in time $O(k\ell)$, whereas potential superior alternatives requiring determination of a mixed Nash equilibrium are PPAD-complete [15]. Second, the pure Nash equilibrium of a two-player zero-sum game corresponds to its max-min strategies (see [51, Proposition 22.2]), making it apt for the class ZEROSUM. Accordingly, a restriction to ZEROSUM is made in the following stability bounds for TSCoEA, which are proven in Appendix B.7 and illustrated in Figure 1.

Theorem 3.8. *Let S_λ be the selection operator for TSCoEA (see Algorithm 3) which uses parameters $k, \ell \in \mathbb{N}$, and let $\gamma \in (0, 1]$.*

¹TSCoEA appears in [31] as Archived Tournament Selection CoEA (ATS-CoEA). As we forego the use of an archive to fit the population-based CoEA with unary variation model, we drop ‘Archived’ from the name.

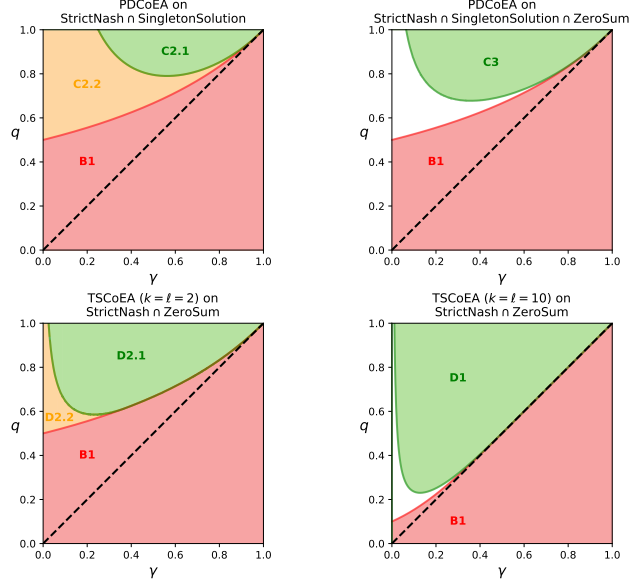


Figure 1: Best proven stability bounds of PDCoEA and TSCoEA on various problem classes. Green regions indicate the algorithm is γ -stable on all instances of the relevant class; red regions indicate the algorithm is not γ -stable on all instances of the relevant class; amber regions indicate there exist some instances in the relevant class on which the algorithm is not γ -stable. For each region, the relevant theoretical result is indicated (e.g., **B1**). As Theorem 3.5 proves the region $q < \gamma$ must always be red, the boundary $q = \gamma$ shows both the improvement made by Theorem 3.6 and also how far the algorithm deviates from the theoretical limit on what could be hoped for in terms of stability.

D1 Let

$$q_0(\gamma) = \frac{\gamma}{(1 - (1 - \gamma)^k)(1 - (1 - \gamma)^\ell)}.$$

If $G \in \text{STRICTNASH} \cap \text{ZEROSUM}$ and $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (q_0(\gamma), 1])$ then $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is γ -stable on G .

D2 Suppose additionally that $k = 2$ and let

$$q_0(\gamma) = \sup_{a \in [\gamma, 1]} \frac{1}{(2 - a)(2 - 2(1 - a)^\ell(1 - \gamma/a) - \gamma/a)}. \quad (7)$$

D2.1 If $G \in \text{STRICTNASH} \cap \text{ZEROSUM}$ and $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (q_0(\gamma), 1])$, then $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is γ -stable on G .

D2.2 There is a sequence of problems $(G_n)_{n=1}^\infty$ with $G_n \in \text{STRICTNASH} \cap \text{ZEROSUM} \cap \text{BITSTRING}_n$ for every $n \in \mathbb{N}$ such that if $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{UNBIASED}(n; [0, q_0(\gamma)))$, then $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is not γ -stable on G_n for n sufficiently large.

4 Empirical analysis of stability

In this section we present the empirical stability of PDCoEA and TSCoEA on a range of zero-sum games and compare our findings to the theoretical bounds derived in Section 3.2. Three problems are considered: Bilinear [30, 32, 40], which emulates a standard game-theoretic saddle point trade-off between two players; PlantedBilinear, a randomly instantiated many-dimensional generalisation of Bilinear; and MBJR_2024 [34], a game based on that first introduced by Maiti et al. [47] as a challenging instance for pairs of no-regret learners. By making minor modifications as necessary, all games have the same search domains $\mathcal{X} = \mathcal{Y} = \{0, 1\}^n$ and have a unique strict Nash equilibrium (x^*, y^*) . These games were chosen because they have large strategy spaces, however the Nash equilibrium is known in each case and so can be injected into the initial population at a controlled rate as necessary to analyse stability. For full descriptions of the three problem instances as well as details of

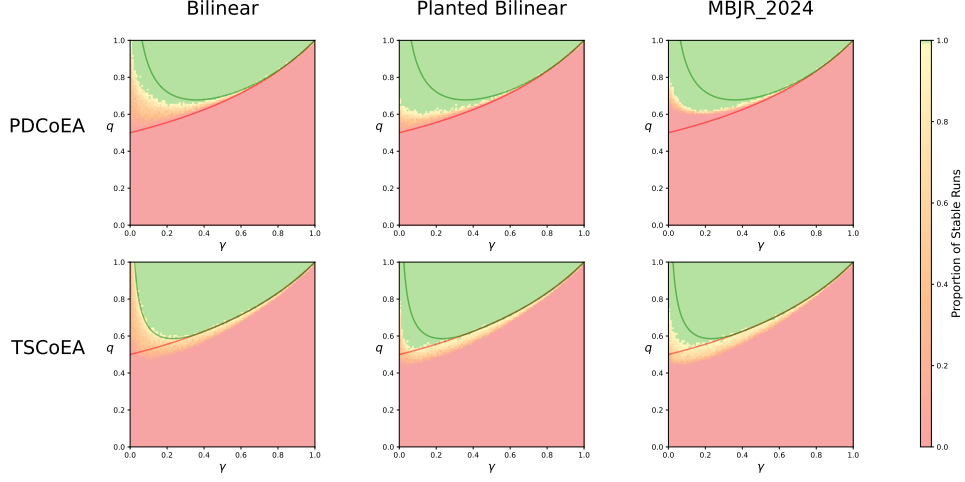


Figure 2: Heatmaps indicating the proportion of runs for which $T_{\text{dep}}^{\gamma}(\mathcal{A}; G) > 4 \times 10^5$ for combinations of $(\gamma, q) \in [0, 1]^2$. Green regions indicate that \mathcal{A} was observed as γ -stable on G for the given q ; points not coloured green indicate that \mathcal{A} was observed to be not γ -stable on G for the given q .

computational resources used to carry out the runs, see Appendix C. Code to reproduce the experiment is available at <https://github.com/asbenford/stability-analysis-of-coeas>.

The mutation operator \mathcal{M} used by each algorithm is standard bitwise mutation, which flips each bit of a searchpoint $x \in \{0, 1\}^n$ independently with probability χ/n for some fixed χ . For a range of combinations of $(\gamma, q) \in [0, 1]^2$, χ was chosen to ensure that $(\mathcal{M}, \mathcal{M}) \in \text{UNBIASED}(n; \{q\})$ and initial populations (P_0, Q_0) of size $\lambda = 10^4$ were generated with $|P_0 \cap \{x^*\}| = \gamma^s$ and $|Q_0 \cap \{y^*\}| = \gamma^{1-s}$ (where $s \sim \text{Unif}([0, 1])$), so that $|(P_0 \times Q_0) \cap \{(x^*, y^*)\}| = \gamma\lambda^2$ (the individuals not initialised on the Nash equilibrium were sampled uniformly at random from $\{0, 1\}^n$). As we cannot test all possible values of δ in Definition 3.2, we make no specific assumption about δ but instead simply test whether $T_{\text{dep}}^{\gamma}(\mathcal{A}; G)$ is small for a large population size. The event that $T_{\text{dep}}^{\gamma}(\mathcal{A}; G)$ is small was then tested for by running the algorithm for 4×10^5 evaluations of the payoff function. By repeating this process 25 times for each (γ, q) -combination, we obtain a broad indication of whether \mathcal{A} is γ -stable on G when using mutation in $\text{UNBIASED}(n; \{q\})$.

The corresponding results are displayed in Figure 2, alongside our best theoretical bounds for stability on the given problems, which are all members of $\text{STRICTNASH} \cap \text{SINGLETONSOLUTION} \cap \text{ZEROSUM}$ (see middle two plots of Figure 1). The most important aspect of each plot is the boundary between the green region indicating stability (as all corresponding runs maintained the initial proportion on Nash) and the non-green region indicating instability. In all cases, the empirical boundary sits between the two theoretical lines, and so the observed stabilities are in close agreement with our theoretical predictions. Of particular note is the fact that for each algorithm the empirical boundary varies depending on problem instance, suggesting that gaps between our theoretical bounds on stability and instability exist due to variation in problem instance and not because potential improvements are missing from the corresponding proofs. In this sense, some of the derived bounds may be best possible.

5 Regret analysis

We now turn our attention to how a deeper understanding of stability of CoEAs can be used to enable regret analysis for coevolution. Let us consider a zero-sum game with unique Nash equilibrium $G = (\mathcal{X}, \mathcal{Y}, f, -f, (x^*, y^*))$. The regret for strategic interaction between two agents is defined with respect to action outputs $(x_t)_{t=0}^{\infty}$ and $(y_t)_{t=0}^{\infty}$ in \mathcal{X} and \mathcal{Y} generated by a learning process. The *cumulative regret* after time T is the total payoff improvement the players could have received, were

they given advanced knowledge of their opponents' actions and allowed to adopt a best (constant) response accordingly. Formally, this is

$$\begin{aligned} R_T &:= \max_{x \in \mathcal{X}} \left(\sum_{t=1}^T (f_1(x, y_t) - f_1(x_t, y_t)) \right) + \max_{y \in \mathcal{Y}} \left(\sum_{t=1}^T (f_2(x_t, y) - f_2(x_t, y_t)) \right) \\ &= \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \left(\sum_{t=1}^T (f(x, y_t) - f(x_t, y)) \right). \end{aligned}$$

The learning process is then said to be ε -Hannan consistent if the time-averaged regret converges almost surely to the interval $[0, \varepsilon]$. That is,

$$\mathbb{P} \left[\limsup_{T \rightarrow \infty} \frac{1}{T} R_T \leq \varepsilon \right] = 1. \quad (8)$$

In the case where $\varepsilon = 0$, the learning process is called *Hannan consistent* or *zero-regret* [11, 37, 56]. It is known that the time-averaged output of a zero-regret algorithm converges to a pure Nash equilibrium when one exists, and ascertaining theoretical guarantees about regret has been fundamental to the development of multi-agent algorithms including Exp3 [3], CFR [64, 9], and FTRL [23]. Such guarantees instill confidence in these algorithms, however they are typically only applicable when the strategy spaces \mathcal{X} and \mathcal{Y} are small enough to query every possible strategy. Here we turn our attention to regret analysis of CoEAs, which potentially apply to games with superpolynomial search spaces where polynomial identification of a pure Nash equilibrium is highly non-trivial.

Assume that at each time step, a population-based CoEA \mathcal{A} outputs a pair (x_t, y_t) of elites selected from the current populations (P_t, Q_t) according to

$$x_t = \operatorname{argmax}_{x \in P_t} \min_{y \in Q_t} f(x, y) \quad y_t = \operatorname{argmin}_{y \in Q_t} \max_{x \in P_t} f(x, y).$$

Recall that for zero-sum games, this policy always ensures $(x_t, y_t) = (x^*, y^*)$ whenever $(x^*, y^*) \in P_t \times Q_t$. Runtime analysis (e.g., [5, 31, 40, 43]) typically proves polynomial bounds on the expected time until (x^*, y^*) is first discovered, however this alone does not ensure a regret guarantee. However, when combined with an added provision of stability (as given by the results of Section 3), it is straightforward to deduce $\varepsilon(\lambda)$ -Hannan consistency, where $\varepsilon(\lambda)$ is a function that decays exponentially in λ . This is handled in the following theorem, which is proven in Appendix B.8. In the statement, the runtime condition is represented using $T_{\text{hit}}^\gamma(\mathcal{A}; G)$ (see Definition 3.1).

Theorem 5.1. *Let $G \in \text{ZEROSUM} \cap \text{SINGLETONSOLUTION}$ and suppose for some $\gamma \in (0, 1]$ and $\tau : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ that the following runtime bound holds for all populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$.*

$$\mathbb{E}[T_{\text{hit}}^\gamma(\mathcal{A}; G) \mid (P_0, Q_0) = (P, Q)] \leq \tau(\lambda). \quad (9)$$

If \mathcal{A} is γ -stable on G then there is a constant c such that \mathcal{A} is $\tau(\lambda)e^{-c\lambda}$ -Hannan consistent.

6 Limitations and further directions

There are several further directions and open questions related to the foundational results on stability of CoEAs set out here. First, the empirical analysis of Section 4 implied that the gaps between the always-stable and never-stable regions in Figure 1 likely occur due to variation in application rather than a lack of tightness of results. This is not yet confirmed rigorously, but could be with the construction and analysis of specific problem instances. Second, there are many important solution concepts in addition to Nash equilibrium, including dominating strategies, evolutionarily stable strategies, and max-min strategies (for non-zero-sum games), and stability analysis can be undertaken using the tools of Section 3.1. We also note that the methodology of the empirical analysis in Section 4 is limited by its applicability only to problems where the solution concept is already known. Finally, and perhaps most importantly, it is noteworthy that the proven regions where γ -stability is assured for general problem classes (C2.1, C3, D1, D2.1) do not meet the line $\gamma = 0$, despite the fact that 0-stability is clearly desirable for ensuring populations will grow on the solution concept for a single individual. While 0-stability is observed for existing algorithms in certain problem instances (see Figure 2), we should strive to design algorithms configurable to assure 0-stability on general problem classes. As our tools provide means to prove whether this criterion is met, designing such algorithms is also a topic for future work.

Acknowledgements

This research was supported by a Turing AI Fellowship (EPSRC grant ref EP/V025562/1). The computations were performed using the University of Birmingham’s BlueBEAR HPC service. See <http://www.birmingham.ac.uk/bear> for more details.

References

- [1] P. J. Angeline and J. B. Pollack. Competitive environments evolve better solutions for complex tasks. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 264–270, 1993.
- [2] K. Arulkumaran, A. Cully, and J. Togelius. AlphaStar: An evolutionary computation perspective. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’19, pages 314–315, 2019.
- [3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
- [4] D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, and T. Graepel. The mechanics of n-player differentiable games. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 354–363, 2018.
- [5] A. Benford and P. K. Lehre. Runtime analysis of coevolutionary algorithms on a class of symmetric zero-sum games. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’24, pages 1542–1550, 2024.
- [6] A. Benford and P. K. Lehre. A general upper bound for the runtime of a coevolutionary algorithm on impartial combinatorial games. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’25, pages 1594–1603, 2025.
- [7] N. Brown, A. Bakhtin, A. Lerer, and Q. Gong. Combining deep reinforcement learning and search for imperfect-information games. In *Advances in Neural Information Processing Systems* 33, NeurIPS ’20, pages 17057–17069, 2020.
- [8] N. Brown, A. Lerer, S. Gross, and T. Sandholm. Deep counterfactual regret minimization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 793–802, 2019.
- [9] N. Brown and T. Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019.
- [10] P. Casgrain, B. Ning, and S. J. and. Deep Q-learning for Nash equilibria: Nash-DQN. *Applied Mathematical Finance*, 29(1):62–78, 2022.
- [11] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [12] E. Chigot and D. G. Wilson. Coevolution of neural networks for agents and environments. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’22, pages 2306–2309, 2022.
- [13] D. Corus, D.-C. Dang, A. V. Eremeev, and P. K. Lehre. Level-based analysis of genetic algorithms and other search processes. *IEEE Transactions on Evolutionary Computation*, 22(5):707–719, 2018.
- [14] V. Costa, N. Lourenço, and P. Machado. Coevolution of generative adversarial networks. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pages 473–487, 2019.
- [15] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

- [16] B. Doerr, C. Doerr, and J. Yang. Optimal parameter choices via precise black-box analysis. *Theoretical Computer Science*, 801:1–34, 2020.
- [17] B. Doerr and F. Neumann. A survey on recent progress in the theory of evolutionary algorithms for discrete optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 1(4):1–43, 2021.
- [18] S. G. Ficici. *Solution concepts in coevolutionary algorithms*. PhD thesis, Brandeis University, 2004.
- [19] A. Gilpin, S. Hoda, J. Peña, and T. Sandholm. Gradient-based algorithms for finding Nash equilibria in extensive form games. In *Internet and Network Economics*, pages 57–69, 2007.
- [20] A. Greenwald and K. Hall. Correlated Q-learning. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML ’03, pages 242–249, 2003.
- [21] R. Guha, R. Mckendrick, B. Feest, and K. Deb. Multi-objective competitive co-evolutionary optimization and regularity-based decision-making for two-agent wargame strategy optimization. *IEEE Transactions on Evolutionary Computation*, 2025.
- [22] A. Hauptman and M. Sipper. GP-EndChess: Using genetic programming to evolve chess endgame players. In *Proceedings of the 8th European Conference on Genetic Programming*, EuroGP ’05, pages 120–131, 2005.
- [23] E. Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3–4):157–325, 2016.
- [24] J. He and X. Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127(1):57–85, 2001.
- [25] J. Heinrich and D. Silver. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*, 2016.
- [26] E. Hemberg, S. Moskal, U.-M. O’Reilly, E. Liu, and L. Fuller. Evolutionary and coevolutionary multi-agent design choices and dynamics. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’25, pages 559–562, 2025.
- [27] E. Hemberg, J. Rosen, G. Warner, S. Wijesinghe, and U.-M. O’Reilly. Detecting tax evasion: a co-evolutionary approach. *Artificial Intelligence and Law*, 24(2):149–182, 2016.
- [28] M. Hevia Fajardo, J. Toutouh, E. Hemberg, U.-M. O’Reilly, and P. K. Lehre. Runtime bounds for a coevolutionary algorithm on classes of potential games. In *Proceedings of the 18th Conference on Foundations of Genetic Algorithms*, FOGA ’25, pages 85–95, 2025.
- [29] M. A. Hevia Fajardo, E. Hemberg, J. Toutouh, U.-M. O’Reilly, and P. K. Lehre. A self-adaptive coevolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’24, pages 841–849, 2024.
- [30] M. A. Hevia Fajardo and P. K. Lehre. How fitness aggregation methods affect the performance of competitive CoEAs on bilinear problems. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’23, pages 1593–1601, 2023.
- [31] M. A. Hevia Fajardo and P. K. Lehre. Ranking diversity benefits coevolutionary algorithms on an intransitive game. In *Parallel Problem Solving from Nature – PPSN XVIII*, pages 213–229, 2024.
- [32] M. A. Hevia Fajardo, P. K. Lehre, and S. Lin. Runtime analysis of a co-evolutionary algorithm: Overcoming negative drift in maximin-optimisation. In *Proceedings of the 17th Conference on Foundations of Genetic Algorithms*, FOGA ’23, pages 73–83, 2023.
- [33] J. Hu and M. P. Wellman. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

- [34] S. Ito, H. Luo, T. Tsuchiya, and Y. Wu. Instance-dependent regret bounds for learning two-player zero-sum games with bandit feedback. In *Proceedings of the 38th Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 2858–2892, 2025.
- [35] W. Jaśkowski, K. Krawiec, and B. Wieloch. Fitnessless coevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’08, pages 355–362, 2008.
- [36] D. Klijn and A. Eiben. A coevolutionary approach to deep multi-agent reinforcement learning. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’21, pages 283–284, 2021.
- [37] V. Kovařík and V. Lisý. Analysis of hannan consistent selection for monte carlo tree search in simultaneous move games. *Machine Learning*, 109(1):1–50, 2020.
- [38] P. K. Lehre. Negative drift in populations. In *Parallel Problem Solving from Nature – PPSN XI*, pages 244–253, 2010.
- [39] P. K. Lehre. Runtime analysis of competitive co-evolutionary algorithms for maximin optimisation of a bilinear function. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’22, pages 1408–1416, 2022.
- [40] P. K. Lehre. Runtime analysis of competitive co-evolutionary algorithms for maximin optimisation of a bilinear function. *Algorithmica*, 86(7):2352–2392, 2024.
- [41] P. K. Lehre, M. Hevia Fajardo, J. Toutouh, E. Hemberg, and U.-M. O’Reilly. Analysis of a pairwise dominance coevolutionary algorithm and DefendIt. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’23, pages 1027–1035, 2023.
- [42] P. K. Lehre and S. Lin. Overcoming binary adversarial optimisation with competitive coevolution. In *Parallel Problem Solving from Nature – PPSN XVIII*, pages 117–132, 2024.
- [43] P. K. Lehre and S. Lin. Towards runtime analysis of population-based co-evolutionary algorithms on sparse binary zero-sum game. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25):27054–27062, 2025.
- [44] P. K. Lehre and C. Witt. Black-box search by unbiased variation. *Algorithmica*, 64(4):623–642, 2012.
- [45] J. Lengler. Drift analysis. In *Theory of Evolutionary Computation: Recent Developments in Discrete Optimization*, pages 89–131. Springer International Publishing, Cham, 2020.
- [46] Y. Liu, A. Zhou, and H. Zhang. Termination detection strategies in evolutionary algorithms: A survey. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’18, pages 1063–1070, 2018.
- [47] A. Maiti, R. Boczar, K. Jamieson, and L. Ratliff. Near-optimal pure exploration in matrix games: A generalization of stochastic bandits and dueling bandits. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 2602–2610, 2024.
- [48] C. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics, 1989: Invited Papers at the Twelfth British Combinatorial Conference*, London Mathematical Society Lecture Note Series, pages 148–188, 1989.
- [49] J. Noble and R. A. Watson. Pareto coevolution: Using performance against coevolved opponents in a game as dimensions for Pareto selection. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, GECCO ’01, pages 493–500, 2001.
- [50] G. Ochoa. Error thresholds in genetic algorithms. *Evolutionary computation*, 14(2):157–182, 2006.
- [51] M. J. Osborne and A. Rubinstein. *A course in game theory*. The MIT Press, 1994.

- [52] T. Paixão, G. Badkobeh, N. Barton, D. Çörüş, D.-C. Dang, T. Friedrich, P. K. Lehre, D. Sudholt, A. M. Sutton, and B. Trubenová. Toward a unifying framework for evolutionary processes. *Journal of Theoretical Biology*, 383:28–43, 2015.
- [53] D. Pfau and O. Vinyals. Connecting generative adversarial networks and actor-critic methods. *arXiv preprint arXiv:1610.01945*, 2016.
- [54] J. B. Pollack and A. D. Blair. Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32(3):225–240, 1998.
- [55] E. Popovici, A. Bucci, R. P. Wiegand, and E. D. De Jong. Coevolutionary principles. In *Handbook of Natural Computing*, pages 987–1033. Springer, 2012.
- [56] R.-J. Qin and Y. Yu. Learning in games: A systematic review. *Science China Information Sciences*, 67(7):171101, 2024.
- [57] F. Schäfer and A. Anandkumar. Competitive gradient descent. In *Advances in Neural Information Processing Systems 32*, NeurIPS ’19, 2019.
- [58] M. Szubert, W. Jaśkowski, and K. Krawiec. On scalability, generalization, and hybridization of coevolutionary learning: A case study for Othello. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(3):214–226, 2013.
- [59] M. J. Turner, E. Hemberg, and U.-M. O’Reilly. Analyzing multi-agent reinforcement learning and coevolution in cybersecurity. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’22, pages 1290–1298, 2022.
- [60] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, Nov 2019.
- [61] R. Wang, J. Lehman, J. Clune, and K. O. Stanley. POET: Open-ended coevolution of environments and their optimized solutions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’19, pages 142–151, 2019.
- [62] K. Xu, C. Li, J. Zhu, and B. Zhang. Understanding and stabilizing GANs’ training dynamics using control theory. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10566–10575, 2020.
- [63] G. Zhang and Y. Yu. Convergence of gradient methods on bilinear zero-sum games. In *International Conference on Learning Representations*, ICLR ’20, 2020.
- [64] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20*, NIPS ’07, 2007.
- [65] A. Żychowski and J. Mańdziuk. Coevolution of players strategies in security games. *Journal of Computational Science*, 68:101980, 2023.

A On the strength of Definition 3.2

Theorem A.1. *Let S_λ be any selection operator, let $G := (\mathcal{X}, \mathcal{Y}, f_1, f_2, A \times B) \in \text{SINGLETONSOLUTION}$, and let $(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (0, 1))$. Then there exists a constant $\bar{\delta} > 0$ such that for all $\lambda \in \mathbb{N}$ and populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$,*

$$\mathbb{P}[T_{\text{dep}}^0(\mathcal{A}; G) > e^{\bar{\delta}\lambda} \mid (P_0, Q_0) = (P, Q)] \leq e^{-\bar{\delta}\lambda},$$

where $\mathcal{A} = \mathcal{A}_S(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}, \lambda)$. In particular, if $h : \mathbb{N} \rightarrow \mathbb{R}$ is any function that grows faster than exponential (that is, $h(\lambda) = \omega(e^{\delta\lambda})$ for any $\delta \in \mathbb{R}$), then $\lim_{\lambda \rightarrow \infty} \mathbb{P}[T_{\text{dep}}^0(\mathcal{A}; G) \leq h(\lambda)] = 1$.

Proof. Write $S = \{(x^*, y^*)\}$. Because

$$\mathcal{M}_{\mathcal{X}}(x)(x) \cdot \mathcal{M}_{\mathcal{Y}}(y)(y) \in (0, 1) \quad (10)$$

holds for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we either have $\mathcal{M}_{\mathcal{X}}(x^*)(x^*) < 1$ or $\mathcal{M}_{\mathcal{Y}}(y^*)(y^*) < 1$. Assume without loss of generality that $\mathcal{M}_{\mathcal{X}}(x^*)(x^*) < 1$ (as the case $\mathcal{M}_{\mathcal{Y}}(y^*)(y^*) < 1$ is similar). Since (10) implies that $\mathcal{M}_{\mathcal{X}}(x)(x) > 0$ for all $x \neq x^*$, we have $\mathcal{M}_{\mathcal{X}}(x)(x^*) < 1$ for all $x \neq x^*$ also. Thus, if we define

$$\bar{\delta} = -2 \min_{x \in \mathcal{X}} \ln(1 - \mathcal{M}_{\mathcal{X}}(x)(x^*)) > 0,$$

it now holds for any $x \in \mathcal{X}$ and $x' \sim \mathcal{M}_{\mathcal{X}}(x)$ that $\mathbb{P}(x' \neq x^*) \geq e^{-\bar{\delta}/2}$. Thus, for any $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$, by considering Line 4 of Algorithm 1,

$$\mathbb{P}((P_{t+1} \times Q_{t+1}) \cap S = \emptyset \mid P_t = P \wedge Q_t = Q) \geq e^{-\bar{\delta}\lambda/2}.$$

We may now bound for any $\tau \in \mathbb{N}$ and $(P, Q) \in \mathcal{X}^\lambda \times \mathcal{Y}^\lambda$,

$$\mathbb{P}[T_{\text{dep}}^0(\mathcal{A}; G) > \tau \mid (P_0, Q_0) = (P, Q)] \leq (1 - e^{-\bar{\delta}\lambda/2})^\tau \leq \exp(-\tau e^{-\bar{\delta}\lambda/2}).$$

Setting $\tau = e^{\bar{\delta}\lambda}$ now yields

$$\mathbb{P}[T_{\text{dep}}^0(\mathcal{A}; G) > e^{\bar{\delta}\lambda} \mid (P_0, Q_0) = (P, Q)] \leq \exp(-e^{\bar{\delta}\lambda/2}) \leq (e^{\bar{\delta}\lambda/2})^{-2} = e^{-\bar{\delta}\lambda},$$

where we have used the identity $\exp(-z) \leq z^{-2}$ for $z > 0$. \square

B Deferred proofs

B.1 Preliminary results

Several of our proofs rely on analysing the distribution of $|(P_{t+1} \times Q_{t+1}) \cap (A \times B)|$ (that is, the number of predator-prey pairs that lie on the solution concept at time $t + 1$) given the value of $|(P_t \times Q_t) \cap (A \times B)|$ is fixed. For this task there are many concentration inequalities that will suffice, however it is particularly convenient to use McDiarmid's inequality [48], given as follows.

Theorem B.1. *Suppose $f : \mathcal{X}_1 \times \dots \times \mathcal{X}_n \rightarrow \mathbb{R}$ has the property that substituting the value of the i^{th} coordinate changes the value of f by at most c_i . Suppose that X_1, \dots, X_n are independent random variables where $X_i \in \mathcal{X}_i$ for each $i \in [n]$. Then, for any $t > 0$,*

$$\begin{aligned} \mathbb{P}(f(X_1, \dots, X_n) \geq \mathbb{E}[f(X_1, \dots, X_n)] + t) &\leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right), \\ \mathbb{P}(f(X_1, \dots, X_n) \leq \mathbb{E}[f(X_1, \dots, X_n)] - t) &\leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right). \end{aligned}$$

In order to later prove conclusion **A2** of Lemma 3.4, roughly speaking we need to estimate the amount of time until the number X_t of predator-prey pairs on the solution concept drops below a certain value. This detail will be handled using the additive drift theorem [24, 45], which provides such estimations provided one can prove a minimum expected drift on such pairs in each generation.

Theorem B.2. *Let $(X_t)_{t=0}^\infty$ be a sequence of non-negative random variables with a finite state space $S \subseteq \mathbb{R}_0^+$ such that $0 \in S$. Let $T := \inf\{t \geq 0 : X_t = 0\}$. Suppose there exists $\delta > 0$ such that for all $s \in S \setminus \{0\}$ and all $t \geq 0$,*

$$\Delta_t(s) := \mathbb{E}[X_t - X_{t+1} \mid X_t = s] \geq \delta.$$

Then

$$\mathbb{E}[T] \leq \frac{\mathbb{E}[X_0]}{\delta}.$$

B.2 Basic calculations

In several of our proofs, certain steps require some calculations that are straightforward, however also lengthy (in many cases they are working through standard calculus arguments to establish the infimum or supremum of certain functions). In order to benefit the flow of presented proofs, we provide these calculations separately here.

Proposition B.3. Let $C \in [0, 3)$ be fixed, and let $\alpha : [0, 1]^2 \rightarrow \mathbb{R}$ and $\beta : [0, 1]^2 \rightarrow \mathbb{R}$ be defined by

$$\begin{aligned}\alpha(a, b) &= a(a + Cb(1 - a)), \\ \beta(a, b) &= b(b + Ca(1 - b)).\end{aligned}$$

It holds for any $\gamma \in (0, 1]$ that

$$\sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)} = ((1 + C)\sqrt{\gamma} - C\gamma)^{-2}.$$

Proof. The case $C = 0$ is trivial, so additionally assume that $C > 0$. Let $g : [\gamma, 1] \rightarrow \mathbb{R}$ be given by $g(a) = \alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)/\gamma$. We have

$$g(a) = \left(a + C\frac{\gamma}{a} - C\gamma\right) \left(\frac{\gamma}{a} + Ca - C\gamma\right) = Ca^2 - (C^2 + C)\gamma a - (C^2 + C)\frac{\gamma^2}{a} + C\frac{\gamma^2}{a^2} + C^2\gamma^2 + \gamma,$$

and hence,

$$\begin{aligned}g'(a) &= 2Ca - (C^2 + C)\gamma + (C^2 + C)\frac{\gamma^2}{a^2} - 2C\frac{\gamma^2}{a^3} \\ &= \frac{C}{a^3}(2a^4 - (1 + C)\gamma a^3 + (1 + C)\gamma^2 a - 2\gamma^2) \\ &= \frac{C}{a^3}(a^2 - \gamma)(2a^2 - (1 + C)\gamma a + 2\gamma) \\ &= \frac{C}{a^3}(a - \sqrt{\gamma})(a + \sqrt{\gamma})(2a^2 - (1 + C)a + 2\gamma).\end{aligned}$$

The discriminant of the quadratic $2a^2 - (1 + C)\gamma a + 2\gamma$ is

$$(1 + C)^2\gamma^2 - 16\gamma \leq (1 + C)^2\gamma^2 - 16\gamma^2 = \gamma^2((1 + C)^2 - 16) < 0.$$

Therefore, $2a^2 - (1 + C)\gamma a + 2\gamma$ is positive for all $a \in \mathbb{R}$. Hence, $h'(a)$ is negative for $a \in [\gamma, \sqrt{\gamma})$ and positive for $a \in (\sqrt{\gamma}, 1]$, and hence

$$\inf_{a \in [\gamma, 1]} g(a) = g(\sqrt{\gamma}) = ((1 + C)\sqrt{\gamma} - C\gamma)^2.$$

The result now follows. \square

Proposition B.4. Let $\alpha : [0, 1]^2 \rightarrow \mathbb{R}$ be defined by $\alpha(a, b) = a(a + 2b(1 - a))$. For all $\gamma \in (0, 9/16]$, we have

$$\inf_{(a, b) \in (0, \sqrt{\gamma}]^2} \frac{ab}{\alpha(a, b) \cdot \alpha(b, a)} \geq (3\sqrt{\gamma} - 2\gamma)^{-2}. \quad (11)$$

Proof. Let $g : (0, \sqrt{\gamma}]^2 \rightarrow \mathbb{R}$ be given by $g(a, b) = \alpha(a, b) \cdot \alpha(b, a)/(ab) = (a + 2b(1 - a))(b + 2a(1 - b))$. We can compute the partial derivative

$$\begin{aligned}\frac{\partial g}{\partial a}(a, b) &= (1 - 2b)(b + 2a - 2ab) + (a + 2b - 2ab)(2 - 2b) \\ &= 5b - 6b^2 + 4a - 12ab + 8ab^2 \\ &= b(5 - 6b) + 4(1 - 2b)(1 - b)a.\end{aligned}$$

If $0 < b \leq 1/2$, then $\frac{\partial g}{\partial a}(a, b) \geq b(5 - 6b) \geq 0$. On the other hand, if $1/2 < b \leq \sqrt{\gamma}$, then $4(1 - 2b)(1 - b)$ is negative, and so

$$\begin{aligned}\frac{\partial g}{\partial a}(a, b) &\geq b(5 - 6b) + 4(1 - 2b)(1 - b)\sqrt{\gamma} \geq b(5 - 6b) + 3(1 - 2b)(1 - b) \\ &= 5b - 6b^2 + 3 - 9b + 6b^2 = 3 - 4b \geq 3 - 4\sqrt{\gamma} \geq 3 - 4(3/4) = 0.\end{aligned}$$

As $\frac{\partial g}{\partial a}$ is non-negative on $(0, \sqrt{\gamma}]^2$, we have $g(a, b) \leq g(\sqrt{\gamma}, b)$ for all $a, b \in (0, \sqrt{\gamma}]$. Therefore, for any $a, b \in (0, \sqrt{\gamma}]^2$, $g(a, b) \leq g(\sqrt{\gamma}, b) = g(b, \sqrt{\gamma}) \leq g(\sqrt{\gamma}, \sqrt{\gamma}) = (3\sqrt{\gamma} - 2\gamma)^2$. Therefore,

$$\inf_{(a,b) \in (0, \sqrt{\gamma}]^2} \frac{ab}{\alpha(a, b) \cdot \alpha(b, a)} = \inf_{(a,b) \in (0, \sqrt{\gamma}]^2} \frac{1}{g(a, b)} \geq \frac{1}{(3\sqrt{\gamma} - 2\gamma)^2},$$

as required. \square

Proposition B.5. Let $\alpha : [0, 1]^2 \rightarrow \mathbb{R}$ and $\beta : [0, 1]^2 \rightarrow \mathbb{R}$ be defined by

$$\begin{aligned}\alpha(a, b) &= (1 - (1 - a)^k)(1 - (1 - b)^\ell), \\ \beta(a, b) &= (1 - (1 - b)^k)(1 - (1 - a)^\ell).\end{aligned}$$

It holds for any $\gamma \in (0, 1]$ that

$$\sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)} = \frac{\gamma}{(1 - (1 - \gamma)^k)(1 - (1 - \gamma)^\ell)}.$$

Proof. Let $g : [\gamma, 1] \rightarrow \mathbb{R}$ be given by $g(a) = \alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)$ and note that $g(a) > 0$ for all $a \in [\gamma, 1]$. We have $g(a) = A(a)B(a)C(a)D(a)$ where

$$\begin{aligned}A(a) &= 1 - (1 - a)^k, & B(a) &= 1 - (1 - \gamma/a)^\ell, \\ C(a) &= 1 - (1 - \gamma/a)^k, & D(a) &= 1 - (1 - a)^\ell.\end{aligned}$$

Therefore we can write

$$\begin{aligned}g'(a) &= g(a) \left(\frac{A'(a)}{A(a)} + \frac{C'(a)}{C(a)} + \frac{D'(a)}{D(a)} + \frac{B'(a)}{B(a)} \right) \\ &= g(a) \left(\frac{k(1 - a)^{k-1}}{1 - (1 - a)^k} - \left(\frac{\gamma}{a^2} \right) \frac{k(1 - \gamma/a)^{k-1}}{1 - (1 - \gamma/a)^k} + \frac{\ell(1 - a)^{\ell-1}}{1 - (1 - a)^\ell} - \left(\frac{\gamma}{a^2} \right) \frac{\ell(1 - \gamma/a)^{\ell-1}}{1 - (1 - \gamma/a)^\ell} \right) \\ &= \frac{g(a)}{a} (h_k(a) - h_k(\gamma/a) + h_\ell(a) - h_\ell(\gamma/a)),\end{aligned}\tag{12}$$

where for $m \in \mathbb{N}$ we define $h_m : [\gamma, 1] \rightarrow \mathbb{R}$ by

$$h_m(a) = \frac{ma(1 - a)^{m-1}}{1 - (1 - a)^m}.$$

We will shortly prove that any $m \in \mathbb{N}$, h_m is a decreasing function. It then follows that $h_k(a) - h_k(\gamma/a) + h_\ell(a) - h_\ell(\gamma/a)$ is decreasing in a . Thus, from (12) it can be seen that $g'(a) \geq 0$ for $a \in [\gamma, \sqrt{\gamma}]$ and $g'(a) \leq 0$ for $a \in [\sqrt{\gamma}, 1]$, and hence

$$\inf_{a \in [\gamma, 1]} g(a) = \min \{g(\gamma), g(1)\} = (1 - (1 - \gamma)^k)(1 - (1 - \gamma)^\ell).$$

The proposition then follows by noting that

$$\sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)} = \frac{\gamma}{\inf_{a \in [\gamma, 1]} g(a)}.$$

Thus all that remains is to prove that h_m is decreasing. To see this, observe using the quotient rule that

$$\begin{aligned}\frac{h'_m(a)}{m} &= \frac{(1 - (1 - a)^m)((1 - a)^{m-1} - (m - 1)a(1 - a)^{m-2}) - a(1 - a)^{m-1}(m(1 - a)^{m-1})}{(1 - (1 - a)^m)^2} \\ &= \frac{(1 - a)^{m-2}[(1 - (1 - a)^m)(1 - a - (m - 1)a) - ma(1 - a)^m]}{(1 - (1 - a)^m)^2} \\ &= \frac{(1 - a)^{m-2}[(1 - (1 - a)^m)(1 - ma) - ma(1 - a)^m]}{(1 - (1 - a)^m)^2} \\ &= \frac{(1 - a)^{m-2}[1 - (1 - a)^m - ma]}{(1 - (1 - a)^m)^2} = \left(\frac{(1 - a)^{m-2}}{(1 - (1 - a)^m)^2} \right) \phi(a)\end{aligned}$$

where $\phi : [0, 1] \rightarrow \mathbb{R}$ is defined as $\phi(a) = 1 - (1 - a)^m - ma$. We have $\phi'(a) = m(1 - a)^{m-1} - m \leq 0$ and $\phi(0) = 0$, and so $\phi(a) \leq 0$ always. Therefore, $h'_m(a) \leq 0$ for all $a \in [\gamma, 1]$, showing that h_m is a decreasing function and completing the proof of the proposition. \square

Proposition B.6. Let $c : [0, 1]^2 \rightarrow \mathbb{R}$ be arbitrary, and let $d : [0, 1]^2 \rightarrow \mathbb{R}$ be such that either $d([0, 1]^2) \subseteq [0, \infty)$ or $d([0, 1]^2) \subseteq (-\infty, 0]$. For $\sigma \in [0, 1]$, let

$$\begin{aligned}\alpha_\sigma(a, b) &= a(c(a, b) + d(a, b) \cdot \sigma), \\ \beta_\sigma(a, b) &= b(c(b, a) + d(b, a) \cdot (1 - \sigma)).\end{aligned}$$

Then, for any $\gamma \in (0, 1]$ it holds that

$$\sup_{a \in [\gamma, 1]} \sup_{\sigma \in [0, 1]} \frac{\gamma}{\alpha_\sigma(a, \gamma/a) \cdot \beta_\sigma(a, \gamma/a)} = \sup_{a \in [\gamma, 1]} \frac{1}{c(a, \gamma/a) \cdot (c(\gamma/a, a) + d(\gamma/a, a))}.$$

Proof. First, note that for any $a, b \in [0, 1]$ and we have

$$\frac{ab}{\alpha_1(b, a) \cdot \beta_1(b, a)} = \frac{ab}{\alpha_0(a, b) \cdot \beta_0(a, b)} = \frac{1}{c(a, b) \cdot (c(b, a) + d(b, a))}. \quad (13)$$

Next, for all fixed $a, b \in (0, 1]$ we have

$$\begin{aligned}\frac{\alpha_\sigma(a, b) \cdot \beta_\sigma(a, b)}{ab} &= (c(a, b) + d(a, b) \cdot \sigma)(c(b, a) + d(b, a) \cdot (1 - \sigma)) \\ &= [c(a, b)(c(b, a) + d(b, a))] + [d(a, b)c(b, a) - c(a, b)d(b, a)] \cdot \sigma \\ &\quad + [d(a, b)d(b, a)] \cdot \sigma(1 - \sigma).\end{aligned}$$

By the assumptions of the proposition, $d(a, b)d(b, a) \geq 0$. Thus, $\alpha_\sigma(a, b) \cdot \beta_\sigma(a, b)$ attains its minimum at some $\sigma \in \{0, 1\}$. We now have

$$\begin{aligned}\sup_{a \in [\gamma, 1]} \sup_{\sigma \in [0, 1]} \frac{\gamma}{\alpha_\sigma(a, \gamma/a) \cdot \beta_\sigma(a, \gamma/a)} &= \sup_{a \in [\gamma, 1]} \max_{\sigma \in \{0, 1\}} \frac{\gamma}{\alpha_\sigma(a, \gamma/a) \cdot \beta_\sigma(a, \gamma/a)} \\ &= \sup_{a \in [\gamma, 1]} \max \left\{ \frac{\gamma}{\alpha_0(a, \gamma/a) \cdot \beta_0(a, \gamma/a)}, \frac{\gamma}{\alpha_1(a, \gamma/a) \cdot \beta_1(a, \gamma/a)} \right\} \\ &\stackrel{(13)}{=} \sup_{a \in [\gamma, 1]} \max \left\{ \frac{\gamma}{\alpha_0(a, \gamma/a) \cdot \beta_0(a, \gamma/a)}, \frac{\gamma}{\alpha_0(\gamma/a, a) \cdot \beta_0(\gamma/a, a)} \right\} \\ &= \max \left\{ \sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha_0(a, \gamma/a) \cdot \beta_0(a, \gamma/a)}, \sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha_0(\gamma/a, a) \cdot \beta_0(\gamma/a, a)} \right\} \\ &= \max \left\{ \sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha_0(a, \gamma/a) \cdot \beta_0(a, \gamma/a)}, \sup_{a' \in [\gamma, 1]} \frac{\gamma}{\alpha_0(a', \gamma/a') \cdot \beta_0(a', \gamma/a')} \right\} \\ &= \sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha_0(a, \gamma/a) \cdot \beta_0(a, \gamma/a)} \\ &\stackrel{(13)}{=} \sup_{a \in [\gamma, 1]} \frac{1}{c(a, \gamma/a) \cdot (c(\gamma/a, a) + d(\gamma/a, a))},\end{aligned}$$

as required. \square

B.3 Proof of Lemma 3.3

Lemma 3.3 is simply the case $|I| = 1$ in the following generalisation.

Lemma B.7. Let S_λ be a selection operator, let $G := (\mathcal{X}, \mathcal{Y}, f_1, f_2, A \times B)$, and let $\gamma \in [0, 1]$. Suppose there exists a family $\{(\alpha_i, \beta_i)\}_{i \in I}$ of pairs of increasing functions $\alpha_i : [0, 1]^2 \rightarrow [0, 1]$ and $\beta_i : [0, 1]^2 \rightarrow [0, 1]$ such that for all $\lambda \in \mathbb{N}$ and populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$, there exists $i \in I$ such that

$$\begin{aligned}\mathbb{P}(x \in A) &\geq \alpha_i \left(\frac{|P \cap A|}{\lambda}, \frac{|Q \cap B|}{\lambda} \right) \\ \mathbb{P}(y \in B) &\geq \beta_i \left(\frac{|P \cap A|}{\lambda}, \frac{|Q \cap B|}{\lambda} \right)\end{aligned}$$

holds whenever $(x, y) \sim \mathcal{S}_\lambda(P, Q)$. Then, provided $(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (q_0(\gamma), 1])$ where

$$q_0(\gamma) := \sup_{a \in [\gamma, 1]} \sup_{i \in I} \frac{\gamma}{\alpha_i(a, \gamma/a) \cdot \beta_i(a, \gamma/a)},$$

$\mathcal{A}_\mathcal{S}(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}, \lambda)$ is γ -stable on G .

Proof. Let $q = \min_{(x, y) \in \mathcal{X} \times \mathcal{Y}} \mathcal{M}_\mathcal{X}(x)(x) \cdot \mathcal{M}_\mathcal{Y}(y)(y)$. Because $(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; (q_0(\gamma), 1])$, we have $q > q_0(\gamma)$. Let $\Lambda_0 \in \mathbb{N}$ and $\varepsilon > 0$ be such that $\gamma + \varepsilon + 2/\Lambda_0 \leq 2$ and

$$q > \sup_{a \in [\gamma, 1]} \sup_{i \in I} \frac{\gamma + \varepsilon + 2/\Lambda_0}{\alpha_i(a, \gamma/a) \cdot \beta_i(a, \gamma/a)}. \quad (14)$$

We remark that ε (and Λ_0) can be chosen to depend only on γ , q , and $\{(\alpha_i, \beta_i)\}_{i \in I}$, and hence have no dependence on λ . Label $S = A \times B$ and let $X_t = |(P_t \times Q_t) \cap S|$. Suppose at time t that $|P_t \cap A| = a\lambda$ and $|Q_t \cap B| = b\lambda$ where $c := ab \geq \gamma$. Using the fact that $b = c/a \geq \gamma/a$ and $a \in [\gamma, 1]$, there exists some $i \in I$ (depending on P_t and Q_t) such that for arbitrary $j \neq k$,

$$\begin{aligned} \mathbb{P}((P_{t+1}(j), Q_{t+1}(k)) \in S) &\geq q \cdot \alpha_i(a, b) \cdot \beta_i(a, b) \geq q \cdot \alpha_i(a, \gamma/a) \cdot \beta_i(a, \gamma/a) \\ &\stackrel{(14)}{\geq} \gamma + \varepsilon + 2/\Lambda_0. \end{aligned} \quad (15)$$

We may write $X_t = f((P_{t+1}(1), Q_{t+1}(1)), \dots, (P_{t+1}(\lambda), Q_{t+1}(\lambda)))$ where

$$f((x_1, y_1), \dots, (x_\lambda, y_\lambda)) = \sum_{j, k \in [\lambda]} \mathbb{1}((x_j, y_k) \in S).$$

Note that modifying the value of any one of the (x_j, y_j) changes the value of $f((x_1, y_1), \dots, (x_\lambda, y_\lambda))$ by at most 2λ . For any $c \geq \gamma$ and $\lambda \geq \Lambda_0$ we have

$$\begin{aligned} \mathbb{E}[X_{t+1} \mid X_t = c\lambda^2] &= \sum_{j, k \in [\lambda]} \mathbb{P}((P_{t+1}(j), Q_{t+1}(k)) \in S) \stackrel{(15)}{\geq} (\gamma + \varepsilon + 2/\Lambda_0)\lambda(\lambda - 1) \\ &= (\gamma + \varepsilon)\lambda^2 + 2\frac{\lambda^2}{\Lambda_0} - (\gamma + \varepsilon + 2/\Lambda_0)\lambda \\ &\geq (\gamma + \varepsilon)\lambda^2 + 2\lambda - 2\lambda = (\gamma + \varepsilon)\lambda^2. \end{aligned}$$

Applying Theorem B.1 now gives

$$\mathbb{P}(X_{t+1} \leq \gamma\lambda^2 \mid X_t = c\lambda^2) \leq \exp\left(-\frac{2\varepsilon^2\lambda^4}{\sum_{j \in [\lambda]} (2\lambda)^2}\right) = \exp(-\tfrac{1}{2}\varepsilon^2\lambda). \quad (16)$$

Write $\mathcal{A} = \mathcal{A}_\mathcal{S}(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}, \lambda)$. For a given τ , if the event $T_{\text{dep}}^\gamma(\mathcal{A}; G) \leq \tau$ occurs then there must exist some t with $0 \leq t < \tau$ such that $X_t \geq \gamma\lambda^2$ and $X_{t+1} \leq \gamma\lambda^2$. Therefore we have the relation of events

$$(T_{\text{dep}}^\gamma(\mathcal{A}; G) \leq \tau) \subseteq \cup_{t < \tau} (X_t \geq \gamma\lambda^2 \wedge X_{t+1} \leq \gamma\lambda^2). \quad (17)$$

By taking $\tau = e^{\frac{1}{4}\varepsilon^2\lambda}$, a union bound now yields for any $(P, Q) \in \mathcal{X}^\lambda \times \mathcal{Y}^\lambda$ satisfying $|(P \times Q) \cap S| \geq \gamma\lambda^2$,

$$\begin{aligned} \mathbb{P}[T_{\text{dep}}^\gamma(\mathcal{A}; G) \leq e^{\frac{1}{4}\varepsilon^2\lambda} \mid (P_0, Q_0) = (P, Q)] &\stackrel{(17)}{\leq} \sum_{t < \tau} \mathbb{P}(X_t \geq \gamma\lambda^2 \wedge X_{t+1} \leq \gamma\lambda^2) \\ &\leq \sum_{t < \tau} \mathbb{P}(X_{t+1} \leq \gamma\lambda^2 \mid X_t \geq \gamma\lambda^2) \\ &\stackrel{(16)}{\leq} \tau \cdot \exp(-\tfrac{1}{2}\varepsilon^2\lambda) = e^{-\frac{1}{4}\varepsilon^2\lambda}, \end{aligned}$$

thus satisfying Definition 3.2 for $\delta = \frac{1}{4}\varepsilon^2$ (which is independent of λ). \square

We now consider a connection between Lemma 3.3 (and implicitly also Lemma B.7) and the coevolutionary level-based theorem [40]. The coevolutionary level-based theorem is stated with respect to a constant γ_0 and subsets of the search domains $A_1, \dots, A_m \subseteq \mathcal{X}$ and $B_1, \dots, B_m \subseteq \mathcal{Y}$, with $A_1 = \mathcal{X}$ and $B_1 = \mathcal{Y}$, such that at each time t , the largest j for which $|(P_t \times Q_t) \cap (A_j \times B_j)| \geq \gamma_0 \lambda^2$ is identified as the ‘current level’. By using assumptions arising from algorithm and problem instance, such as lower bounds on the probability of generating offspring on the next level, the theorem derives an upper bound on the expected time for a CoEA to reach the final level $A_m \times B_m$ (which is typically identified with the solution concept). This form of analysis extends the classical level-based theorem, which uses similar constructions to derive upper bounds on expected runtime for EAs [13].

One of the assumptions needed to apply the coevolutionary level-based theorem ((G2b) in [40, Theorem 3]) is that if j is the current level (so that $|(P_t \times Q_t) \cap (A_j \times B_j)| \geq \gamma_0 \lambda$), then the expected proportion of level- j pairs in the next generation is strictly larger than γ_0 . Precisely, the condition is

$$\mathbb{P}(P_{t+1}(k) \in A_j) \cdot \mathbb{P}(Q_{t+1}(\ell) \in B_j) \geq (1 + \delta)\gamma_0,$$

where δ is a constant (on which the eventual runtime bound depends) and $k, \ell \in [\lambda]$ are arbitrary indices. In the proof of the coevolutionary level-based theorem, this assumption is used to show the populations do not regress to a lower level, or informally that they have stabilised on the current level. It is therefore not too surprising that if we identify the solution concept $A \times B$ as the current level (and also identify $\gamma_0 = \gamma$), then the assumptions of Lemma 3.3 imply assumption (G2b). Indeed, with the assumptions of Lemma 3.3, setting $a = |P_t \cap A|$ and arguing as in (15) yields

$$\begin{aligned} \mathbb{P}(P_{t+1}(k) \in A) \cdot \mathbb{P}(Q_{t+1}(\ell) \in B) &\geq q \cdot \alpha(a, b) \cdot \beta(a, b) \geq q \cdot \alpha(a, \gamma/a) \cdot \beta(a, \gamma/a) \\ &\geq \gamma + \varepsilon = (1 + \varepsilon/\gamma)\gamma = (1 + \delta)\gamma_0, \end{aligned}$$

where ε is chosen (depending on the mutation operators) as in (14) and we set $\delta = \varepsilon/\gamma_0$. Thus, there is potential for much of the analysis in this paper (particularly the derivations of the functions α and β within the proofs of Theorem 3.7 and Theorem 3.8) to also be used to derive level-based runtime results for specific problems and problem classes.

B.4 Proof of Lemma 3.4

Before proving Lemma 3.4, we will state and prove the general result outlining situations where instability occurs which applies to games with arbitrary search domains rather than just bitstrings. Lemma 3.4 will then be derived using this result.

In general, if a mutation operator is somehow rigged to produce offspring inside or near to a game’s solution concept with unreasonable probability, then stability can be trivially guaranteed even for CoEAs using impractical selection operators (for example, a CoEA using a mutation operator that returns elements of the solution concept with probability 1 will always be γ -stable for any $\gamma \in [0, 1]$). Therefore, to obtain useful negative results it is necessary to have an underlying assumption related to the innate tendency the mutation operator has for moving towards or into the solution concept. In the case of Lemma 3.4 such an assumption is implicit from the fact that the relevant mutation operators over $\{0, 1\}^n$ are unbiased (see Definition 2.3). As there is not an analogous notion of unbiasedness for arbitrary domains, for the more general lemma in this appendix we will simply bound above the probability of mutating into the solution concept from outside (see (19)).

Lemma B.8. *Let $q, \gamma, \zeta \in [0, 1]$, let S_λ be a selection operator, and let $\mathcal{G} \subseteq \text{SINGLETONSOLUTION}$ be a problem class. Suppose there exist continuous increasing functions $\alpha : [0, 1]^2 \rightarrow [0, 1]$ and $\beta : [0, 1]^2 \rightarrow [0, 1]$ such that for all $\lambda \in \mathbb{N}$, problems $(\mathcal{X}, \mathcal{Y}, f_1, f_2, (x^*, y^*)) \in \mathcal{G}$, and populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$,*

$$\mathbb{P}(x = x^*) \leq \alpha\left(\frac{|P \cap \{x^*\}|}{\lambda}, \frac{|Q \cap \{y^*\}|}{\lambda}\right) \quad \text{and} \quad \mathbb{P}(y = y^*) \leq \beta\left(\frac{|P \cap \{x^*\}|}{\lambda}, \frac{|Q \cap \{y^*\}|}{\lambda}\right)$$

holds whenever $(x, y) \sim S_\lambda(P, Q)$. Then provided $(\mathcal{M}_\mathcal{X}, \mathcal{M}_\mathcal{Y}) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; [0, q])$ where

$$q < \sup_{a \in [\gamma, 1]} \frac{\gamma - \zeta}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)}, \quad (18)$$

and also

$$\max \left\{ \sup_{x \neq x^*} \mathcal{M}_\mathcal{X}(x)(x^*), \sup_{y \neq y^*} \mathcal{M}_\mathcal{Y}(y)(y^*) \right\} \leq \zeta/3, \quad (19)$$

the algorithm $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is not γ -stable on any $G \in \mathcal{G}$.

Proof. Let $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{MUTATION}(\mathcal{X}, \mathcal{Y}; [0, q])$ and define

$$q_X = \sup_{x \in \mathcal{X}} \mathcal{M}_X(x)(x) \quad \text{and} \quad q_Y = \sup_{y \in \mathcal{Y}} \mathcal{M}_Y(y)(y). \quad (20)$$

Note that by Definition 2.2 we have $q_X q_Y \leq q$. Adopting the notation of a given selection and mutation step in Algorithm 1 (that is, sampling $(x, y) \sim \mathcal{S}_\lambda(P_t, Q_t)$ and then sampling $P_{t+1}(j) \sim \mathcal{M}_X(x)$ and $Q_{t+1}(j) \sim \mathcal{M}_Y(y)$), for fixed P_t and Q_t we can compute

$$\begin{aligned} \mathbb{P}(P_{t+1}(j) = x^*) &= \mathbb{P}(P_{t+1}(j) = x^* \mid x = x^*)\mathbb{P}(x = x^*) + \mathbb{P}(P_{t+1}(j) = x^* \mid x \neq x^*)\mathbb{P}(x \neq x^*) \\ &\stackrel{(20)}{\leq} q_X \cdot \alpha \left(\frac{|P_t \cap \{x^*\}|}{\lambda}, \frac{|Q_t \cap \{y^*\}|}{\lambda} \right) + \mathbb{P}(P_{t+1}(j) = x^* \mid x \neq x^*) \\ &\leq q_X \cdot \alpha \left(\frac{|P_t \cap \{x^*\}|}{\lambda}, \frac{|Q_t \cap \{y^*\}|}{\lambda} \right) + \sup_{x \neq x^*} \mathcal{M}_X(x)(x^*) \\ &\stackrel{(19)}{\leq} q_X \cdot \alpha \left(\frac{|P_t \cap \{x^*\}|}{\lambda}, \frac{|Q_t \cap \{y^*\}|}{\lambda} \right) + \frac{\zeta}{3}. \end{aligned}$$

Similar calculations apply for $Q_{t+1}(j)$, and so we have for arbitrary $j \in [\lambda]$,

$$\begin{aligned} \mathbb{P}(P_{t+1}(j) = x^*) &\leq q_X \cdot \alpha \left(\frac{|P_t \cap \{x^*\}|}{\lambda}, \frac{|Q_t \cap \{y^*\}|}{\lambda} \right) + \frac{\zeta}{3}, \\ \mathbb{P}(Q_{t+1}(j) = y^*) &\leq q_Y \cdot \beta \left(\frac{|P_t \cap \{x^*\}|}{\lambda}, \frac{|Q_t \cap \{y^*\}|}{\lambda} \right) + \frac{\zeta}{3}. \end{aligned} \quad (21)$$

Using (18), let $a \in [\gamma, 1]$ be chosen such that, setting $b = \gamma/a$, we have

$$q < \frac{\gamma - \zeta}{\alpha(a, b) \cdot \beta(a, b)}.$$

Using that α and β are continuous, let $\eta > 0$ and $\Lambda \in \mathbb{N}$ be chosen such that

$$q < \frac{\gamma - \eta - \frac{1}{\Lambda} - \zeta}{\alpha(a + \frac{1}{\Lambda}, b + \frac{1}{\Lambda}) \cdot \beta(a + \frac{1}{\Lambda}, b + \frac{1}{\Lambda})}. \quad (22)$$

Let $\lambda \geq \Lambda$, $G \in \mathcal{G}$, and consider $\mathcal{A} := \mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ with initial populations $P_0 \in \mathcal{X}^\lambda$ and $Q_0 \in \mathcal{Y}^\lambda$ chosen to satisfy $|P_0| = \lceil a\lambda \rceil$ and $|Q_0| = \lceil b\lambda \rceil$. Using that $\lceil a\lambda \rceil \leq a + \frac{1}{\Lambda}$ and $\lceil b\lambda \rceil \leq b + \frac{1}{\Lambda}$ together with the fact that α and β are increasing, it holds for $j \neq k$ that

$$\begin{aligned} \mathbb{P}((P_1(j), Q_1(k)) \in S) &\stackrel{(21)}{\leq} (q_X \cdot \alpha(a + \frac{1}{\Lambda}, b + \frac{1}{\Lambda}) + \frac{\zeta}{3})(q_Y \cdot \beta(a + \frac{1}{\Lambda}, b + \frac{1}{\Lambda}) + \frac{\zeta}{3}) \\ &\leq q \cdot \alpha(a + \frac{1}{\Lambda}, b + \frac{1}{\Lambda}) \cdot \beta(a + \frac{1}{\Lambda}, b + \frac{1}{\Lambda}) + \zeta \\ &\stackrel{(22)}{\leq} \gamma - \eta - \frac{1}{\Lambda} \leq \gamma - \eta - \frac{1}{\lambda}. \end{aligned} \quad (23)$$

Setting $S = \{(x^*, y^*)\}$, we may write $|(P_1 \times Q_1) \cap S| = f((P_1(1), Q_1(1)), \dots, (P_1(\lambda), Q_1(\lambda)))$ where

$$f((x_1, y_1), \dots, (x_\lambda, y_\lambda)) = \sum_{j, k \in [\lambda]} \mathbb{1}((x_j, y_k) \in S).$$

Note that modifying the value of (x_j, y_j) changes the value of $f((x_1, y_1), \dots, (x_\lambda, y_\lambda))$ by at most 2λ . We also have

$$\mathbb{E}[|(P_1 \times Q_1) \cap S|] = \sum_{j, k \in [\lambda]} \mathbb{P}((P_1(j), Q_1(k)) \in S) \stackrel{(23)}{\leq} (\gamma - \eta - 1/\lambda)\lambda^2 + \lambda = (\gamma - \eta)\lambda^2.$$

Applying Theorem B.1 now gives,

$$\mathbb{P}[T_{\text{dep}}^\gamma(\mathcal{A}; G) = 1] \geq \mathbb{P}[|(P_1 \times Q_1) \cap S| \leq \gamma\lambda^2] \leq \exp\left(-\frac{2\eta^2\lambda^4}{\sum_{j \in [\lambda]} (2\lambda)^2}\right) = \exp(-\tfrac{1}{2}\eta^2\lambda),$$

and hence \mathcal{A} is not γ -stable on G . \square

We are now ready to derive Lemma 3.4 using the more general result.

Proof of Lemma 3.4. First we will derive **A1**. Using (4), let $n_0 \in \mathbb{N}$ be chosen such that

$$q < \sup_{a \in [\gamma, 1]} \frac{\gamma - 3/n_0}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)}. \quad (24)$$

Suppose that $n \geq n_0$, $G \in \mathcal{G}_n$, and $(\mathcal{M}_X, \mathcal{M}_Y) \in \text{UNBIASED}(n; [0, q])$, as per the conditions of **A1**. In accordance with the characterisation of Lemma 1 of [16], an unbiased mutation can be simulated by choosing a random number $r \in \{0, 1, \dots, n\}$ and then flipping a uniformly selected subset of r bits. Thus, when combined with our minor assumption that $\mathcal{M}_X(\mathbf{0}^n)(\mathbf{1}^n) \leq 1/n$, we have that $\mathcal{M}_X(x)(x') = \mathcal{M}_Y(x)(x') \leq 1/n \leq 1/n_0$ holds for every $x, x' \in \{0, 1\}^n$ with $x \neq x'$. Therefore, condition (19) of Lemma B.8 also holds for $\zeta = 3/n_0$. Furthermore, the condition (18) of Lemma B.8 also holds for $\zeta = 3/n_0$ due to (24). Therefore, Lemma B.8 shows that $\mathcal{A}_S(\mathcal{M}_X, \mathcal{M}_Y, \lambda)$ is not γ -stable on any $G \in \mathcal{G}_n$, and so **A1** holds.

We now turn to proving **A2**. Let $\delta > 0$ now be chosen such that

$$q < \inf_{(a,b) \in [0, \sqrt{\gamma}]^2} \frac{(1 - 2\delta)ab}{\alpha(a, b) \cdot \alpha(b, a)}. \quad (25)$$

Note that the assumption in Definition 2.3 that $\mathcal{M}_Y = \mathcal{M}_X$ implies that $q_X = q_Y \leq \sqrt{q}$ (where q_X and q_Y are as defined in (20)). Therefore, as the calculations at the beginning of the proof of Lemma B.8 apply with $\zeta = 3/n$, we can deduce from (21) that for arbitrary $j \in [\lambda]$,

$$\begin{aligned} \mathbb{P}(P_{t+1}(j) \in A \mid \mathcal{F}_t) &\leq \sqrt{q} \cdot \alpha \left(\frac{|P_t \cap \{x^*\}|}{\lambda}, \frac{|Q_t \cap \{y^*\}|}{\lambda} \right) + \frac{1}{n}, \\ \mathbb{P}(Q_{t+1}(j) \in B \mid \mathcal{F}_t) &\leq \sqrt{q} \cdot \beta \left(\frac{|P_t \cap \{x^*\}|}{\lambda}, \frac{|Q_t \cap \{y^*\}|}{\lambda} \right) + \frac{1}{n}. \end{aligned} \quad (26)$$

We will examine the random variable

$$X_t = \begin{cases} |(P_t \times Q_t) \cap S| & \text{if } |P_t \cap A| \leq \sqrt{\gamma}\lambda \text{ and } |Q_t \cap B| \leq \sqrt{\gamma}\lambda, \\ 0 & \text{otherwise.} \end{cases}$$

Suppose that $X_t = s \geq \varepsilon(n)\lambda^2$. Writing $|P_t \cap \{x^*\}| = a\lambda$ and $|Q_t \cap \{y^*\}| = b\lambda$, we have $a, b \leq \sqrt{\gamma}$ and $ab \geq \varepsilon(n)$. We have for $j \neq k$,

$$\begin{aligned} \mathbb{P}((P_{t+1}(j), Q_{t+1}(k)) = (x^*, y^*)) &\stackrel{(26)}{\leq} (\sqrt{q} \cdot \alpha(a, b) + \frac{1}{n})(\sqrt{q} \cdot \alpha(b, a) + \frac{1}{n}) \\ &\leq q \cdot \alpha(a, b) \cdot \alpha(b, a) + \frac{3}{n} \\ &\stackrel{(25)}{\leq} (1 - 2\delta)ab + \frac{3}{n}, \end{aligned}$$

and for $j = k$,

$$\mathbb{P}((P_{t+1}(j), Q_{t+1}(j)) = (x^*, y^*)) \leq Kab + \frac{2}{n}.$$

We now bound

$$\begin{aligned} \Delta_t(s) &:= \mathbb{E}[X_t - X_{t+1} \mid X_t = s] \geq \mathbb{E}[X_t - |(P_{t+1} \times Q_{t+1}) \cap S| \mid X_t = s] \\ &= ab\lambda^2 - \sum_{j,k \in [\lambda]} \mathbb{P}((P_{t+1}(j), Q_{t+1}(k)) \in S \mid X_t = s) \\ &\geq ab\lambda^2 - (1 - 2\delta)ab\lambda^2 - (3/n)\lambda^2 - Kab\lambda - (2/n)\lambda \\ &\geq ((2\delta - K/\lambda)ab - 5/n)\lambda^2 \geq (\delta\varepsilon(n) - 5/n)\lambda^2 \\ &\geq \left(\frac{\delta\varepsilon(n)}{2} \right) \lambda^2. \end{aligned}$$

Thus, by applying the additive drift theorem (Theorem B.2), and also noting that $X_0 \leq \lambda^2$, we have for $T = \inf \{t : X_t \leq \varepsilon(n)\lambda^2\}$,

$$\mathbb{E}[T] \leq \frac{2}{\delta\varepsilon(n)}.$$

(We remark that Theorem B.2 is not applied directly to the stochastic process $(X_t)_{t=0}^\infty$, but rather to a modified process $\tilde{X}_t := \max\{X_t - \varepsilon(n)\lambda^2, 0\}$; the fact that $\mathbb{E}[\tilde{X}_0] \leq \lambda^2$ is also used.) By Markov's inequality, it therefore holds for any $\tau(n) \geq 1$ that

$$\mathbb{P}[T \geq \tau] \leq \frac{2}{\delta \varepsilon(n) \tau(n)}.$$

Next, let $T_{\text{bad}} = \min\{t : \max\{|P_t \cap \{x^*\}|, |Q_t \cap \{y^*\}|\} > \sqrt{\gamma}\lambda\}$. If $|P_t \cap \{x^*\}| = a\lambda \in [0, \sqrt{\gamma}\lambda]$ and $|Q_t \cap \{y^*\}| = b\lambda \in [0, \sqrt{\gamma}\lambda]$, then by recalling that $n \geq 2/(\delta\sqrt{\gamma})$,

$$\begin{aligned} \mathbb{E}[|P_{t+1} \cap \{x^*\}|] &\leq (\sqrt{q} \cdot \alpha(a, b) + \tfrac{1}{n})\lambda \leq (\sqrt{q} \cdot \alpha(\sqrt{\gamma}, \sqrt{\gamma}) + \tfrac{1}{n})\lambda \\ &\leq (\sqrt{(1-2\delta)\gamma} + \tfrac{1}{n})\lambda \leq ((1-\delta)\sqrt{\gamma} + \tfrac{1}{n})\lambda \leq (1-\delta/2)\sqrt{\gamma}\lambda. \end{aligned}$$

Using a similar calculation for $\mathbb{E}[|Q_{t+1} \cap \{y^*\}|]$, applying Theorem B.1 and taking a union bound over time steps, we now obtain

$$\mathbb{P}[T_{\text{bad}} \leq \tau(n)] \leq 2\tau(n) \cdot \exp\left(-\frac{2\delta^2\gamma\lambda^2/4}{\lambda}\right) = 2\tau(n) \cdot e^{-\delta^2\gamma\lambda/2} \leq 2\tau(n) \cdot \frac{1}{\delta\tau(n)^2\varepsilon(n)}.$$

Setting $S = \{(x^*, y^*)\}$, we have $T = \min\{T_{\text{dep}}^{(n)}(S), T_{\text{bad}}\}$, and hence

$$\mathbb{P}[T_{\text{dep}}^{(n)}(S) \geq \tau(n)] \leq \mathbb{P}[T \geq \tau(n)] + \mathbb{P}[T_{\text{bad}} \leq \tau(n)] \leq \frac{3}{\delta\tau(n)\varepsilon(n)},$$

as required. \square

B.5 Proof of Theorem 3.6

While its exact statement is not necessary for the proof of Theorem 3.6, the following is a formalisation of what it means to be a k -candidate selection operator. The proof of Theorem 3.6 is then given afterwards.

Definition B.9. A k -candidate selection operator is a selection operator \mathcal{S}_λ for which there exists a function $\bar{\mathcal{S}}_\lambda : \mathcal{X}^k \times \mathcal{Y}^k \times \mathcal{X}^\lambda \times \mathcal{Y}^\lambda \rightarrow \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ satisfying the following two properties.

E1 For all $P' \in \mathcal{X}^k$, $Q' \in \mathcal{Y}^k$, $P \in \mathcal{X}^\lambda$, and $Q \in \mathcal{Y}^\lambda$,

$$\mathbb{P}(x \in P' \wedge y \in Q') = 1$$

holds whenever $(x, y) \sim \bar{\mathcal{S}}_\lambda(P', Q', P, Q)$.

E2 For all $P := (x_1, \dots, x_\lambda) \in \mathcal{X}^\lambda$ and $Q := (y_1, \dots, y_\lambda) \in \mathcal{Y}^\lambda$,

$$\frac{1}{\lambda^{2k}} \sum_{i_1, \dots, i_k \in [\lambda]} \sum_{j_1, \dots, j_k \in [\lambda]} \bar{\mathcal{S}}_\lambda((x_{i_1}, \dots, x_{i_k}), (y_{j_1}, \dots, y_{j_k}), P, Q) = \mathcal{S}_\lambda(P, Q).$$

Proof of Theorem 3.6. Let $G \in \text{BITSTRING}_n \cap \text{SINGLETONSOLUTION}$ and let (x^*, y^*) be the solution concept for G . Let $\lambda \in \mathbb{N}$ and let $P \in \mathcal{X}^\lambda$, $Q \in \mathcal{Y}^\lambda$ be arbitrary. Write $a = |P \cap \{x^*\}|/\lambda$ and $b = |Q \cap \{y^*\}|/\lambda$. Because \mathcal{S}_λ is a k -candidate selection operator, if $(x, y) \sim \mathcal{S}_\lambda(P, Q)$ then the event $x = x^*$ can occur only at least one of the k predator candidates is equal to x^* , an event which occurs with probability $1 - (1-a)^k$. In particular $\mathbb{P}(x = x^*) \leq 1 - (1-a)^k$. More formally, if we write $P = (x_1, \dots, x_\lambda) \in \mathcal{X}^\lambda$ and $Q = (y_1, \dots, y_\lambda) \in \mathcal{Y}^\lambda$, and also define $I = \{i \in [\lambda] : x_i \neq x^*\}$

(so that $|I| = (1 - a)\lambda$), we have

$$\begin{aligned}
\mathbb{P}(x = x^*) &= \sum_{y \in \mathcal{Y}} \mathcal{S}_\lambda(P, Q)(x, y) \\
&\stackrel{\mathbf{E2}}{=} \sum_{y \in \mathcal{Y}} \frac{1}{\lambda^{2k}} \sum_{i_1, \dots, i_k \in [\lambda]} \sum_{j_1, \dots, j_k \in [\lambda]} \bar{\mathcal{S}}_\lambda((x_{i_1}, \dots, x_{i_k}), (y_{j_1}, \dots, y_{j_k}), P, Q)(x, y) \\
&\stackrel{\mathbf{E1}}{=} \sum_{y \in \mathcal{Y}} \frac{1}{\lambda^{2k}} \sum_{(i_1, \dots, i_k) \in [\lambda]^k \setminus I^k} \sum_{j_1, \dots, j_k \in [\lambda]} \bar{\mathcal{S}}_\lambda((x_{i_1}, \dots, x_{i_k}), (y_{j_1}, \dots, y_{j_k}), P, Q)(x, y) \\
&= \frac{1}{\lambda^{2k}} \sum_{(i_1, \dots, i_k) \in [\lambda]^k \setminus I^k} \sum_{j_1, \dots, j_k \in [\lambda]} \sum_{y \in \mathcal{Y}} \bar{\mathcal{S}}_\lambda((x_{i_1}, \dots, x_{i_k}), (y_{j_1}, \dots, y_{j_k}), P, Q)(x, y) \\
&\leq \frac{1}{\lambda^{2k}} \sum_{(i_1, \dots, i_k) \in [\lambda]^k \setminus I^k} \sum_{j_1, \dots, j_k \in [\lambda]} 1 = \frac{\lambda^k(\lambda^k - ((1 - a)\lambda)^k)}{\lambda^{2k}} = 1 - (1 - a)^k.
\end{aligned}$$

Similarly, $\mathbb{P}(y = y^*) \leq 1 - (1 - b)^k$. Moreover, the event $x = x^* \wedge y = y^*$ can occur only if at least one of the k predator candidates is equal to x^* and one of the prey candidates is equal to y^* , which occurs with probability at most $(1 - (1 - a)^k)(1 - (1 - b)^k) \leq k^2 ab$. Therefore, the prerequisites of Lemma 3.4 are satisfied with $\alpha(a, b) = 1 - (1 - a)^k$, $\beta = 1 - (1 - b)^k$, and $K = k^2$. **B1** now follows by applying **A1** and verifying that

$$\sup_{a \in [\gamma, 1]} \frac{\gamma}{(1 - (1 - a)^k)(1 - (1 - \gamma/a)^k)} = \frac{\gamma}{1 - (1 - \gamma)^k}.$$

A2 follows by applying **A2** and verifying that

$$\inf_{(a, b) \in (0, \sqrt{\gamma}]^2} \frac{ab}{(1 - (1 - a)^k)(1 - (1 - b)^k)} \geq \inf_{(a, b) \in (0, \sqrt{\gamma}]^2} \frac{ab}{k^2 ab} = \frac{1}{k^2}.$$

□

B.6 Proof of Theorem 3.7

Proof of Theorem 3.7. We prove each of part of the theorem in turn.

C1: Let $G \in \text{STRICTNASH}$ and let $A \times B$ be the solution concept for G . Recall that $f_1(x, y) \geq f_1(x', y)$ and $f_2(x, y) \geq f_2(x, y')$ holds for all $x \in A$, $x' \notin A$, $y \in B$, and $y' \notin B$. Let $\lambda \in \mathbb{N}$ and $P \in \mathcal{X}^\lambda$, $Q \in \mathcal{Y}^\lambda$ be arbitrary. Write $a = |P \cap A|/\lambda$ and $b = |Q \cap B|/\lambda$. If $(x, y) \sim \mathcal{S}_\lambda(P, Q)$, then letting x_1, x_2 be independent uniform samples from P and y_1, y_2 be independent uniform samples from Q , we have

$$\begin{aligned}
\mathbb{P}(x \in A) &\geq \mathbb{P}(x_1 \in A \wedge x_2 \in A) + \frac{1}{2} \mathbb{P}(x_1 \in A \wedge y_1 \in B \wedge x_2 \notin A) \\
&\quad + \mathbb{P}(x_1 \notin A \wedge x_2 \in A \wedge y_2 \in B) \\
&= a^2 + \frac{3}{2} ab(1 - a) = a(a + \frac{3}{2} b(1 - a)).
\end{aligned}$$

Similarly, $\mathbb{P}(y \in B) \geq b(b + \frac{3}{2} a(1 - b))$. Let $\alpha : [0, 1]^2 \rightarrow \mathbb{R}$ and $\beta : [0, 1]^2 \rightarrow \mathbb{R}$ be given by $\alpha(a, b) = a(a + \frac{3}{2} b(1 - a))$ and $\beta(a, b) = b(b + \frac{3}{2} a(1 - b))$. **C1** now follows from Lemma 3.3 by verifying that

$$\sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)} = (\frac{5}{2} \sqrt{\gamma} - \frac{3}{2} \gamma)^2,$$

which can be seen by setting $C = 3/2$ in Proposition B.3.

C2: Let $G \in \text{STRICTNASH} \cap \text{SINGLETONSOLUTION}$ and let $A \times B$ be the solution concept for G . Recall that $f_1(x, y) \geq f_1(x', y)$ and $f_2(x, y) \geq f_2(x, y')$ holds for all $x \in A$, $x' \notin A$, $y \in B$, and $y' \notin B$. Let $\lambda \in \mathbb{N}$ and $P \in \mathcal{X}^\lambda$, $Q \in \mathcal{Y}^\lambda$ be arbitrary. Write $a = |P \cap A|/\lambda$ and $b = |Q \cap B|/\lambda$. If $(x, y) \sim \mathcal{S}_\lambda(P, Q)$, then letting x_1, x_2 be independent uniform samples from P and y_1, y_2 be independent uniform samples from Q , we have

$$\begin{aligned}
\mathbb{P}(x \in A) &\geq \mathbb{P}(x_1 \in A \wedge x_2 \in A) + \mathbb{P}(x_1 \in A \wedge y_1 \in B \wedge x_2 \notin A) \\
&\quad + \mathbb{P}(x_1 \notin A \wedge x_2 \in A \wedge y_2 \in B) \\
&= a^2 + 2ab(1 - a) = a(a + 2b(1 - a)).
\end{aligned}$$

Similarly, $\mathbb{P}(y \in B) \geq b(b + 2a(1 - b))$. Let $\alpha : [0, 1]^2 \rightarrow [0, 1]$ and $\beta : [0, 1]^2 \rightarrow [0, 1]$ be given by $\alpha(a, b) = a(a + 2b(1 - a))$ and $\beta(a, b) = b(b + 2a(1 - b))$. **C2.1** now follows from Lemma 3.3 by verifying that

$$\sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)} = (3\sqrt{\gamma} - 2\gamma)^{-2}, \quad (27)$$

which can be seen by setting $C = 2$ in Proposition B.3. Next, let $G_n = (\{0, 1\}^n, \{0, 1\}^n, f_1, f_2, (\mathbf{1}^n, \mathbf{1}^n))$, where f_1 and f_2 are given by

$$f_1(x, y) = \begin{cases} 0 & \text{if } x \neq \mathbf{1}^n \text{ and } y = \mathbf{1}^n, \\ 1 & \text{if } x = \mathbf{1}^n \text{ and } y \neq \mathbf{1}^n, \\ 2 & \text{otherwise,} \end{cases}$$

$$f_2(x, y) = \begin{cases} 1 & \text{if } x \neq \mathbf{1}^n \text{ and } y = \mathbf{1}^n, \\ 0 & \text{if } x = \mathbf{1}^n \text{ and } y \neq \mathbf{1}^n, \\ 2 & \text{otherwise,} \end{cases}$$

and label $A = B = \{\mathbf{1}^n\}$. Let $\lambda \in \mathbb{N}$ and $P \in \mathcal{X}^\lambda, Q \in \mathcal{Y}^\lambda$ be arbitrary. Write $a = |P \cap A|/\lambda$ and $b = |Q \cap B|/\lambda$. If $(x, y) \sim \mathcal{S}_\lambda(P, Q)$, then letting x_1, x_2 be independent uniform samples from P and y_1, y_2 be independent uniform samples from Q , we have

$$\begin{aligned} \mathbb{P}(x \in A) &= \mathbb{P}(x_1 \in A \wedge x_2 \in A) + \mathbb{P}(x_1 \in A \wedge y_1 \in B \wedge x_2 \notin A) \\ &\quad + \mathbb{P}(x_1 \notin A \wedge x_2 \in A \wedge y_2 \in B) \\ &= a^2 + 2ab(1 - a) = a(a + 2b(1 - a)). \end{aligned}$$

Similarly,

$$\mathbb{P}(y \in B) = b(b + 2a(1 - b)).$$

Using also the fact that \mathcal{S}_λ is a 2-candidate selection operator, the prerequisites of Lemma 3.4 are satisfied with $\alpha(a, b) = a(a + 2b(1 - a))$, $\beta(a, b) = b(b + 2a(1 - b))$, and $K = 4$ (this is covered in detail in the proof of Theorem 3.6 provided in Appendix B.5). **C2.2** now follows by applying **A1** using (27). Because $K < \infty$ and $\alpha(a, b) = \beta(b, a)$, in order to additionally deduce **C2.3** from **A2**, we only need to prove that for all $\gamma \in (0, 9/16]$,

$$\inf_{(a, b) \in (0, \sqrt{\gamma}]^2} \frac{ab}{\alpha(a, b) \cdot \alpha(b, a)} \geq (3\sqrt{\gamma} - 2\gamma)^{-2}.$$

This is confirmed by Proposition B.4.

C3: Let $G \in \text{STRICTNASH} \cap \text{SINGLETONSOLUTION} \cap \text{ZEROSUM}$ have payoff function f and solution concept (x^*, y^*) . Let $\lambda \in \mathbb{N}$ and $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ be arbitrary. Write $a = |P \cap \{x^*\}|/\lambda$ and $b = |Q \cap \{y^*\}|/\lambda$. Moreover, let $r = (1 - a)\lambda$ and $s = (1 - b)\lambda$, and note that by reordering the elements of P and Q if necessary, we may assume that $P(j) \neq x^*$ whenever $j \in [r]$ and $Q(k) \neq y^*$ whenever $k \in [s]$. Given $j \in [r]$ we will write

$$p_j = \frac{|\{k \in [s] : f(P(j), Q(k)) < f(x^*, Q(k))\}|}{s},$$

and given $k \in [s]$ we will write

$$q_k = \frac{|\{j \in [r] : f(P(j), Q(k)) < f(P(j), y^*)\}|}{r},$$

Note that $\frac{1}{r} \sum_{j \in [r]} p_j$ is the proportion of pairs $(x, y) \in (P \setminus \{x^*\}) \times (Q \setminus \{y^*\})$ satisfying $f(x, y) < f(x^*, y)$, and that $\frac{1}{s} \sum_{k \in [s]} q_k$ is the proportion of pairs $(x, y) \in (P \setminus \{x^*\}) \times (Q \setminus \{y^*\})$ satisfying $f(x, y) > f(x, y^*)$. Because $f(x, y^*) > f(x^*, y^*) > f(x, y^*)$ holds for any $(x, y) \in (P \setminus \{x^*\}) \times (Q \setminus \{y^*\})$, all such pairs satisfy at least one of these conditions, and hence

$$\frac{1}{r} \sum_{j \in [r]} p_j + \frac{1}{s} \sum_{k \in [s]} q_k \geq 1. \quad (28)$$

If $(x, y) \sim \mathcal{S}_\lambda(P, Q)$, then letting x_1, x_2 be independent uniform samples from P and y_1, y_2 be independent uniform samples from Q it is not difficult to see that

$$\begin{aligned}\mathbb{P}(x \in A \mid x_1 \in A \wedge x_2 \notin A \wedge y_1 \in B \wedge y_2 \in B) &= 1, \\ \mathbb{P}(x \in A \mid x_1 \in A \wedge x_2 \notin A \wedge y_1 \in B \wedge y_2 \notin B) &= 1, \\ \mathbb{P}(x \in A \mid x_1 \in A \wedge x_2 \notin A \wedge y_1 \notin B \wedge y_2 \in B) &\geq 0.\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}(x \in A \mid x_1 \notin A \wedge x_2 \in A \wedge y_1 \in B \wedge y_2 \in B) &= 1, \\ \mathbb{P}(x \in A \mid x_1 \notin A \wedge x_2 \in A \wedge y_1 \in B \wedge y_2 \notin B) &= 1, \\ \mathbb{P}(x \in A \mid x_1 \notin A \wedge x_2 \in A \wedge y_1 \notin B \wedge y_2 \in B) &= 1.\end{aligned}$$

There are two remaining conditioning cases that require more care: $E_1 := x_1 \in A \wedge x_2 \notin A \wedge y_1 \notin B \wedge y_2 \notin B$ and $E_2 := x_1 \notin A \wedge x_2 \in A \wedge y_1 \notin B \wedge y_2 \notin B$. To examine these, write $C_j = \{k \in [s] : f(P(j), Q(k)) < f(x^*, Q(k))\}$ so that $p_j = |C_j|/r$, and write $Q(C_j) = \{Q(k) : k \in C_j\}$. We now have

$$\begin{aligned}\mathbb{P}(x \in A \mid E_1) &\geq \frac{1}{r} \sum_{j \in [r]} \mathbb{P}(y_1 \in Q(C_j) \wedge y_2 \in Q(C_j) \wedge f(P(j), y_1) \leq f(P(j), y_2) \mid E_1) \\ &\geq \frac{1}{r} \sum_{j \in [r]} \frac{1}{2} \mathbb{P}(y_1 \in Q(C_j) \wedge y_2 \in Q(C_j) \mid E_1) = \frac{1}{r} \sum_{j \in [r]} \frac{1}{2} p_j^2.\end{aligned}$$

Additionally,

$$\mathbb{P}(x \in A \mid E_2) \geq \mathbb{P}(f(x_1, y_1) < f(x^*, y_1) \mid E_2) \geq \frac{1}{r} \sum_{j \in [r]} \mathbb{P}(y_1 \in Q(C_j) \mid E_2) = \frac{1}{r} \sum_{j \in [r]} p_j$$

Combining all of these observations using the law of total probability,

$$\begin{aligned}\mathbb{P}(x \in A) &\geq a^2 + a(1-a) \left(b^2 + 3b(1-b) + (1-b)^2 \frac{1}{r} \sum_{j \in [r]} \frac{1}{2} p_j^2 + b^2 + (1-b)^2 \frac{1}{r} \sum_{j \in [r]} p_j \right) \\ &= a^2 + a(1-a) \left(3b - b^2 + (1-b)^2 \frac{1}{r} \sum_{j \in [r]} \left(\frac{1}{2} p_j^2 + p_j \right) \right) \\ &\geq a(a + (1-a) \left(3b - b^2 + (1-b)^2 \left(\frac{1}{2} \left(\frac{1}{r} \sum_{j \in [r]} p_j \right)^2 + \frac{1}{r} \sum_{j \in [r]} p_j \right) \right)),\end{aligned}$$

where in the final line we have used Jensen's inequality. A similar argument for y yields.

$$\mathbb{P}(y \in B) \geq b(b + (1-b) \left(3a - a^2 + (1-a)^2 \left(\frac{1}{2} \left(\frac{1}{s} \sum_{k \in [s]} q_k \right)^2 + \frac{1}{s} \sum_{k \in [s]} q_k \right) \right)).$$

Let $\{(\alpha_\sigma, \beta_\sigma)\}_{\sigma \in [0,1]}$ be the family of pairs of increasing functions $\alpha_\sigma : [0,1]^2 \rightarrow [0,1]$ and $\beta_\sigma : [0,1]^2 \rightarrow [0,1]$ given by

$$\begin{aligned}\alpha_\sigma(a, b) &= a(a + (1-a)(3b - b^2 + (1-b)^2 \sigma(1 + \frac{1}{2}\sigma))), \\ \beta_\sigma(a, b) &= b(b + (1-b)(3a - a^2 + (1-a)^2 (1-\sigma)(1 + \frac{1}{2}(1-\sigma)))).\end{aligned}$$

It now holds (in part due to (28)) that for all populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ there exists $\sigma \in [0, 1]$ such that

$$\begin{aligned}\mathbb{P}(x \in A) &\geq \alpha_\sigma \left(\frac{|P \cap A|}{\lambda}, \frac{|Q \cap B|}{\lambda} \right) \\ \mathbb{P}(y \in B) &\geq \beta_\sigma \left(\frac{|P \cap A|}{\lambda}, \frac{|Q \cap B|}{\lambda} \right)\end{aligned}$$

holds whenever $(x, y) \sim \mathcal{S}_\lambda(P, Q)$. The result now follows from Lemma B.7. \square

B.7 Proof of Theorem 3.8

The following simple lemma will be useful in the proof of Theorem 3.8 for identifying conditions for TSCoEA to select parents on the solution concept of a game in $\text{STRICTNASH} \cap \text{ZEROSUM}$. The proof of Theorem 3.8 is then given afterwards.

Lemma B.10. *Let $G \in \text{STRICTNASH} \cap \text{ZEROSUM}$ have payoff function f and solution concept $A \times B$. Suppose $X \subseteq \mathcal{X}$ and $Y \subseteq \mathcal{Y}$ are such that $X \cap A \neq \emptyset$ and $Y \cap B \neq \emptyset$. Then $\mathbb{P}(\arg\max_{x \in X} \min_{y \in Y} f(x, y) \in A) = 1$ and $\mathbb{P}(\arg\max_{y \in Y} \min_{x \in X} (-f(x, y)) \in B) = 1$.*

Proof. Let $x^* \in X \cap A$ be arbitrary. Using that $G \in \text{STRICTNASH} \cap \text{ZEROSUM}$, we have $f(x^*, y) < f(x^*, y')$ for all $y \in B$ and $y' \notin B$. In particular, there is some $y^* \in Y \cap B$ such that

$$\min_{y \in Y} f(x^*, y) = f(x^*, y^*) \quad (29)$$

Using again that $G \in \text{STRICTNASH} \cap \text{ZEROSUM}$, we have

$$f(x, y^*) > f(x', y^*) \quad \text{for all } x \in A \text{ and } x' \notin A. \quad (30)$$

Thus, for any $x' \in X \setminus A$,

$$\min_{y \in Y} f(x', y) \leq f(x', y^*) \stackrel{(30)}{<} f(x^*, y^*) \stackrel{(29)}{=} \min_{y \in Y} f(x^*, y),$$

and hence we have $\mathbb{P}(\arg\max_{x \in X} \min_{y \in Y} f(x, y) \in A) = 1$. The corresponding proof that $\mathbb{P}(\arg\max_{y \in Y} \min_{x \in X} (-f(x, y)) \in B) = 1$ is similar. \square

Proof of Theorem 3.8. We prove each of part of the theorem in turn.

D1: Let $G \in \text{STRICTNASH} \cap \text{ZEROSUM}$ and let $A \times B$ be the solution concept for G . Let $\lambda \in \mathbb{N}$ and $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ be arbitrary. Write $a = |P \cap A|/\lambda$ and $b = |Q \cap B|/\lambda$. If $(x, y) \sim \mathcal{S}_\lambda(P, Q)$, then letting x_1, \dots, x_k be independent uniform samples from P and y_1, \dots, y_ℓ be independent uniform samples from Q we have (using Lemma B.10)

$$\begin{aligned} \mathbb{P}(x \in A) &\geq \mathbb{P}(\{x_1, \dots, x_k\} \cap A \neq \emptyset \wedge \{y_1, \dots, y_\ell\} \cap B \neq \emptyset) \\ &= (1 - (1 - a)^k)(1 - (1 - b)^\ell). \end{aligned}$$

Similarly, $\mathbb{P}(y \in B) \geq (1 - (1 - b)^k)(1 - (1 - a)^\ell)$. Let $\alpha : [0, 1]^2 \rightarrow \mathbb{R}$ and $\beta : [0, 1]^2 \rightarrow \mathbb{R}$ be given by $\alpha(a, b) = (1 - (1 - a)^k)(1 - (1 - b)^\ell)$ and $\beta(a, b) = (1 - (1 - b)^k)(1 - (1 - a)^\ell)$. **D1** now follows from Lemma 3.3 by verifying that

$$\sup_{a \in [\gamma, 1]} \frac{\gamma}{\alpha(a, \gamma/a) \cdot \beta(a, \gamma/a)} = \frac{\gamma}{(1 - (1 - \gamma)^k)(1 - (1 - \gamma)^\ell)},$$

which is confirmed by Proposition B.5.

D2.1: Let $G \in \text{STRICTNASH} \cap \text{ZEROSUM}$ have payoff function f and solution concept $A \times B$. Let us write

$$f_X = \min_{x \in A} \min_{y \notin B} f(x, y) \quad f_Y = \max_{y \in B} \max_{x \notin A} f(x, y). \quad (31)$$

Note that if $(x^*, y) \in A \times (\mathcal{Y} \setminus B)$ and $(x, y^*) \in (\mathcal{X} \setminus A) \times B$, then because $G \in \text{STRICTNASH} \cap \text{ZEROSUM}$,

$$f(x, y^*) < f(x^*, y^*) < f(x^*, y).$$

Therefore, $f_X > f_Y$.

Let $\lambda \in \mathbb{N}$ and $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ be arbitrary. Write $a = |P \cap A|/\lambda$ and $b = |Q \cap B|/\lambda$. Moreover, let $r = (1 - a)\lambda$ and $s = (1 - b)\lambda$, and note that by reordering the elements of P and Q if necessary, we may assume that $P(j) \notin A$ whenever $j \in [r]$ and $Q(k) \notin B$ whenever $k \in [s]$. Given $j \in [r]$ we will write

$$p_j = \frac{|\{k \in [s] : f(P(j), Q(k)) \geq f_X\}|}{s},$$

and given $k \in [s]$ we will write

$$q_k = \frac{|\{j \in [r] : f(P(j), Q(k)) \leq f_Y\}|}{r},$$

Note that $\frac{1}{r} \sum_{j \in [r]} p_j$ is the proportion of pairs $(x, y) \in (P \setminus A) \times (Q \setminus B)$ satisfying $f(x, y) \geq f_X$, and that $\frac{1}{s} \sum_{k \in [s]} q_k$ is the proportion of pairs $(x, y) \in (P \setminus A) \times (Q \setminus B)$ satisfying $f(x, y) \leq f_Y$. Because $f_X > f_Y$, these are disjoint sets, and hence

$$\frac{1}{r} \sum_{j \in [r]} p_j + \frac{1}{s} \sum_{k \in [s]} q_k \leq 1. \quad (32)$$

If $(x, y) \sim \mathcal{S}_\lambda(P, Q)$, then letting x_1, x_2 be independent uniform samples from P and y_1, \dots, y_ℓ be independent uniform samples from Q we have

$$\begin{aligned} \mathbb{P}(x \in A) &= \mathbb{P}(x_1 \in A \wedge x_2 \in A) + \mathbb{P}(x \in A \mid x_1 \in A \wedge x_2 \notin A) \cdot \mathbb{P}(x_1 \in A \wedge x_2 \notin A) \\ &\quad + \mathbb{P}(x \in A \mid x_1 \notin A \wedge x_2 \in A) \cdot \mathbb{P}(x_1 \notin A \wedge x_2 \in A) \\ &= a^2 + a(1-a) \cdot (\mathbb{P}(x \in A \mid x_1 \in A \wedge x_2 \notin A) + \mathbb{P}(x \in A \mid x_1 \notin A \wedge x_2 \in A)) \\ &= a^2 + 2a(1-a) \cdot \mathbb{P}(x \in A \mid x_1 \in A \wedge x_2 \notin A). \end{aligned}$$

Let E_1 be the event $x_1 \in A \wedge x_2 \notin A$ and let E_2 be the event $y_1, \dots, y_\ell \notin B$. Because E_1 and E_2 are independent events, we have $\mathbb{P}(E_2 \mid E_1) = \mathbb{P}(E_2) = (1-b)^\ell$ and hence

$$\begin{aligned} \mathbb{P}(x \in A \mid E_1) &= \mathbb{P}(x \in A \mid E_1 \wedge E_2) \cdot \mathbb{P}(E_2 \mid E_1) + \mathbb{P}(x \in A \mid E_1 \wedge \overline{E_2}) \cdot \mathbb{P}(\overline{E_2} \mid E_1) \\ &= \mathbb{P}(x \in A \mid E_1 \wedge E_2) \cdot (1-b)^\ell + \mathbb{P}(x \in A \wedge E_1 \wedge \overline{E_2}) \cdot (1 - (1-b)^\ell). \end{aligned}$$

Using Lemma B.10,

$$\mathbb{P}(x \in A \mid E_1 \wedge \overline{E_2}) = \mathbb{P}(x \in A \mid x_1 \in A \wedge x_2 \notin A \wedge \{y_1, \dots, y_\ell\} \cap B \neq \emptyset) = 1.$$

Additionally,

$$\begin{aligned} \mathbb{P}(x \notin A \mid E_1 \wedge E_2) &= \mathbb{P}(x \notin A \mid x_1 \in A \wedge x_2 \notin A \wedge y_1, \dots, y_\ell \notin B) \\ &\leq \mathbb{P}(\min_{i \in [\ell]} f(x_2, y_i) \geq \min_{i \in [\ell]} f(x_1, y_i) \mid x_1 \in A \wedge x_2 \notin A \wedge y_1, \dots, y_\ell \notin B) \\ &\stackrel{(31)}{\leq} \mathbb{P}(\min_{i \in [\ell]} f(x_2, y_i) \geq f_X \mid x_1 \in A \wedge x_2 \notin A \wedge y_1, \dots, y_\ell \notin B) \\ &= \mathbb{P}(\min_{i \in [\ell]} f(x_2, y_i) \geq f_X \mid x_2 \notin A \wedge y_1, \dots, y_\ell \notin B) \\ &= \frac{1}{r} \sum_{j \in [r]} \mathbb{P}(\min_{i \in [\ell]} f(x_2, y_i) \geq f_X \mid x_2 = P(j) \wedge y_1, \dots, y_\ell \notin B) \\ &= \frac{1}{r} \sum_{j \in [r]} \mathbb{P}(\min_{i \in [\ell]} f(P(j), y_i) \geq f_X \mid y_1, \dots, y_\ell \notin B) \\ &= \frac{1}{r} \sum_{j \in [r]} p_j^\ell \leq \frac{1}{r} \sum_{j \in [r]} p_j. \end{aligned}$$

Combining these observations, we have

$$\begin{aligned} \mathbb{P}(x \in A) &\geq a^2 + 2a(1-a) \cdot [(1-b)^\ell \cdot (1 - \frac{1}{r} \sum_{j \in [r]} p_j) + (1 - (1-b)^\ell)] \\ &= a^2 + 2a(1-a)(1 - (1-b)^\ell) \cdot \frac{1}{r} \sum_{j \in [r]} p_j \\ &= a(a + 2(1-a)(1 - (1-b)^\ell) \cdot \frac{1}{r} \sum_{j \in [r]} p_j) \\ &= a(2 - a - 2(1-a)(1-b)^\ell) \cdot \frac{1}{r} \sum_{j \in [r]} p_j \end{aligned}$$

A similar argument for y yields

$$\mathbb{P}(y \in B) \geq b(2 - b - 2(1-b)(1-a)^\ell) \cdot \frac{1}{s} \sum_{k \in [s]} q_k.$$

Let $\{(\alpha_\sigma, \beta_\sigma)\}_{\sigma \in [0,1]}$ be the family of pairs of increasing functions $\alpha_\sigma : [0,1]^2 \rightarrow [0,1]$ and $\beta_\sigma : [0,1]^2 \rightarrow [0,1]$ given by

$$\begin{aligned}\alpha_\sigma(a, b) &= a(2 - a - 2(1 - a)(1 - b)^\ell \sigma), \\ \beta_\sigma(a, b) &= b(2 - b - 2(1 - b)(1 - a)^\ell (1 - \sigma)).\end{aligned}$$

It now holds (in part due to (32)) that for all populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ there exists $\sigma \in [0, 1]$ such that

$$\begin{aligned}\mathbb{P}(x \in A) &\geq \alpha_\sigma\left(\frac{|P \cap A|}{\lambda}, \frac{|Q \cap B|}{\lambda}\right) \\ \mathbb{P}(y \in B) &\geq \beta_\sigma\left(\frac{|P \cap A|}{\lambda}, \frac{|Q \cap B|}{\lambda}\right)\end{aligned}$$

holds whenever $(x, y) \sim \mathcal{S}_\lambda(P, Q)$. Thus, the result now follows from Lemma B.7 by using Proposition B.6 to establish that

$$\sup_{a \in [\gamma, 1]} \sup_{\sigma \in [0, 1]} \frac{\gamma}{\alpha_\sigma(a, \gamma/a) \cdot \beta_\sigma(a, \gamma/a)} = \sup_{a \in [\gamma, 1]} \frac{1}{(2 - a)(2 - 2(1 - a)^\ell (1 - \gamma/a) - \gamma/a)}.$$

D2.2: Let $G_n = (\{0, 1\}^n, \{0, 1\}^n, f, -f, (\mathbf{1}^n, \mathbf{1}^n))$, where f is given by

$$f(x, y) = \begin{cases} 0 & \text{if } x = \mathbf{1}^n \text{ and } y = \mathbf{0}^n, \\ -1 & \text{if } x \neq \mathbf{1}^n \text{ and } y = \mathbf{1}^n, \\ 1 & \text{if } x = \mathbf{1}^n \text{ and } y \neq \mathbf{1}^n, \\ -2 & \text{otherwise,} \end{cases}$$

label $A = B = \{\mathbf{1}^n\}$. Let $\lambda \in \mathbb{N}$ and $P \in \mathcal{X}^\lambda$, $Q \in \mathcal{Y}^\lambda$ be arbitrary. Write $a = |P \cap A|/\lambda$ and $b = |Q \cap B|/\lambda$. If $(x, y) \sim \mathcal{S}_\lambda(P, Q)$, then letting x_1, x_2 be independent uniform samples from P and y_1, y_2 be independent uniform samples from Q , we have

$$\mathbb{P}(x \in A) = \mathbb{P}(x_1 \in A \vee x_2 \in A) = 2a - a^2 = a(2 - a).$$

Additionally,

$$\begin{aligned}\mathbb{P}(y \in B) &= \mathbb{P}(y_1 \in B \wedge y_2 \in B) + \mathbb{P}(y_1 \in B \wedge y_2 \notin B \wedge \{x_1, x_2\} \cap A \neq \emptyset) \\ &\quad + \mathbb{P}(y_1 \notin B \wedge y_2 \in B \wedge \{x_1, x_2\} \cap A \neq \emptyset) \\ &= b^2 + 2b(1 - b)(1 - (1 - a)^2) = b(b + 2(1 - b)(1 - (1 - a)^2)) \\ &= b(2 - 2(1 - a)^2(1 - b) - b).\end{aligned}$$

Using also the fact that \mathcal{S}_λ is a 2-candidate selection operator, the prerequisites of Lemma 3.4 are now satisfied with $\alpha(a, b) = a(2 - a)$, $\beta(a, b) = b(2 - 2(1 - a)^2(1 - b) - b)$, and $K = 4$ (this is covered in detail in the proof of Theorem 3.6 provided in Appendix B.5). **D2.2** now follows by applying **A1**. \square

B.8 Proof of Theorem 5.1

Proof of Theorem 5.1. Using that \mathcal{A} is γ -stable on G , let $\delta > 0$ be a constant such that, for all populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$ satisfying $|(P \times Q) \cap \{(x^*, y^*)\}| \geq \gamma\lambda^2$, it holds that

$$\mathbb{P}[T_{\text{dep}}^\gamma(\mathcal{A}; G) \leq e^{2\delta\lambda} \mid (P_0, Q_0) = (P, Q)] \leq e^{-2\delta\lambda}. \quad (33)$$

Set $a = \lfloor e^{2\delta\lambda} \rfloor$. Given $t \in \mathbb{N}$, let G_t be the event that $(x^*, y^*) \in P_t \times Q_t$. For $i \in \mathbb{N}$, let us define the time interval $I_i = [ia] \setminus [(i-1)a]$, so that $|I_i| = a$. Define

$$T_i = \min \{a + 1\} \cup \{t \in [a] : G_{(i-1)a+t} \text{ holds}\}.$$

From (9), we have $\mathbb{E}[T_i] \leq \tau(\lambda)$.

Let E_i denote the event that $(x_t, y_t) = (x^*, y^*)$ for all but at most $e^{\delta\lambda}$ values of $t \in I_i$. By (33) the probability that G_t holds for all $(i-1)a + T_i \leq t \leq ia$ is at least $(1 - e^{-2\delta\lambda})$. Therefore, for all populations $P \in \mathcal{X}^\lambda$ and $Q \in \mathcal{Y}^\lambda$,

$$\begin{aligned} \mathbb{P}(E_i \mid (P_{(i-1)a}, Q_{(i-1)a}) = (P, Q)) &\geq \mathbb{P}(T_i \leq e^{\delta\lambda}) \cdot (1 - e^{-2\delta\lambda}) \\ &\geq (1 - e^{-\delta\lambda} \mathbb{E}[T_i])(1 - e^{-2\delta\lambda}) \geq 1 - 2\tau(\lambda)e^{-\delta\lambda}. \end{aligned} \quad (34)$$

Let X_1, X_2, \dots be a sequence of independent identically distributed random variables with

$$\begin{aligned} \mathbb{P}(X_i = a) &= 2\tau(\lambda)e^{-\delta\lambda}, \\ \mathbb{P}(X_i = e^{\delta\lambda}) &= 1 - 2\tau(\lambda)e^{-\delta\lambda}. \end{aligned}$$

For $i \in \mathbb{N}$, let $Y_i = |\{t \in I_i : G_t \text{ does not hold}\}|$. Using (34), it holds for all $i \in \mathbb{N}$ and $y_1, \dots, y_{i-1} \in \{0\} \cup [a]$ that

$$\mathbb{P}(Y_i \leq e^{\delta\lambda} \mid Y_1 = y_1 \wedge \dots \wedge Y_{i-1} = y_{i-1}) \geq 1 - 2\tau(\lambda)e^{-\delta\lambda}.$$

In particular, if $\bar{X} = \limsup_{k \rightarrow \infty} \frac{1}{k} \sum_{i \in [k]} X_i$ and $\bar{Y} = \limsup_{k \rightarrow \infty} \frac{1}{k} \sum_{i \in [k]} Y_i$, then

$$\bar{X} \succcurlyeq \bar{Y} \quad (35)$$

We now turn our attention to (8). Writing

$$F := \max_{(x,y) \in \mathcal{X} \times \mathcal{Y}} f(x, y) - \min_{(x,y) \in \mathcal{X} \times \mathcal{Y}} f(x, y),$$

We first note that for any $x \in X$ and $y \in Y$,

$$f(x, y_t) - f(x_t, y) \leq F(1 - \mathbf{1}(G_t)). \quad (36)$$

Thus for any T ,

$$\begin{aligned} \frac{1}{T} \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \left(\sum_{t=1}^T (f(x, y_t) - f(x_t, y)) \right) &\leq \frac{F}{T} \sum_{t=1}^T (1 - \mathbf{1}(G_t)) \leq \frac{aF}{T} + \frac{F}{T} \sum_{t=1}^{\lfloor T/a \rfloor} (1 - \mathbf{1}(G_t)) \\ &\leq \frac{aF}{T} + \frac{F}{a \lfloor T/a \rfloor} \sum_{t=1}^{\lfloor T/a \rfloor} (1 - \mathbf{1}(G_t)) \\ &= \frac{aF}{T} + \frac{F}{a \lfloor T/a \rfloor} \sum_{i=1}^{\lfloor T/a \rfloor} Y_i, \end{aligned}$$

and hence,

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \left(\sum_{t=1}^T (f(x, y_t) - f(x_t, y)) \right) \leq \frac{F}{a} \limsup_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k Y_i = \frac{F}{a} \bar{Y}. \quad (37)$$

Additionally, by the strong law of large numbers,

$$\mathbb{P}(\bar{X} \leq \mathbb{E}[X_1]) = 1. \quad (38)$$

Next, note that there is a constant c (depending on F and δ) such that

$$\begin{aligned} \frac{F}{a} \mathbb{E}[X_1] &= \frac{F}{a} (2\tau(\lambda)e^{-\delta\lambda}a + e^{\delta\lambda} - 2\tau(\lambda)) \geq \tau(\lambda) \cdot 2F(e^{-\delta\lambda} - 1/a) \\ &= \tau(\lambda) \cdot 2F(e^{-\delta\lambda} - 1/\lfloor e^{2\delta\lambda} \rfloor) \geq \tau(\lambda)e^{-c\lambda}. \end{aligned} \quad (39)$$

We now have

$$\begin{aligned} \mathbb{P} \left[\limsup_{T \rightarrow \infty} \frac{1}{T} \max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \left(\sum_{t=1}^T (f(x, y_t) - f(x_t, y)) \right) \leq \tau(\lambda)e^{-c\lambda} \right] &\stackrel{(37)}{\geq} \mathbb{P} \left[\frac{F}{a} \bar{Y} \leq \tau(\lambda)e^{-c\lambda} \right] \\ &\stackrel{(35)}{\geq} \mathbb{P} \left[\frac{F}{a} \bar{X} \leq \tau(\lambda)e^{-c\lambda} \right] \stackrel{(39)}{\geq} \mathbb{P} \left[\frac{F}{a} \bar{X} \leq \frac{F}{a} \mathbb{E}[X_1] \right] = \mathbb{P}(\bar{X} \leq \mathbb{E}[X_1]) \stackrel{(38)}{=} 1 \end{aligned}$$

Thus, \mathcal{A} is $\tau(\lambda)e^{-c\lambda}$ -Hannan consistent, as required. \square

C Experimental details

The runs used to produce the data represented in Figure 2 were executed on an internal cluster provisioned with 1344 CPU cores and 6TB of RAM and had a wall-clock time of 36 hours, resulting in a maximum total provision of 48384 core-hours. A Python implementation of the experiment is publicly available at <https://github.com/asbenford/stability-analysis-of-coeas>.

We now provide complete definitions for the three problems considered in the empirical analysis of Section 4. As all three problems are zero-sum games use the same action spaces $\mathcal{X} = \mathcal{Y} = \{0, 1\}^n$, it suffices in each case to specify the payoff function f for player 1 (as we then take $f_1 = f$ and $f_2 = -f$). In all cases, the problems were instantiated with $n = 50$. In the subsequent definitions we write $|x|$ for the number of 1-bits in $x \in \{0, 1\}^n$ (that is, $|x| = \sum_{i \in [n]} x(i)$).

C.1 Bilinear

Let $a, b \in \{0, \dots, n\}$. Given $x \in \{0, 1\}^n$, write $\text{LO}(x)$ for the length of a maximal prefix of 1-bits of x and write $\text{TZ}(x)$ for the length of a maximal suffix of 0-bits of x . Formally,

$$\begin{aligned}\text{LO}(x) &= \sum_{i \in [n]} \prod_{j \in [i]} x(j), \\ \text{TZ}(x) &= \sum_{i \in [n]} \prod_{j \in [i]} (1 - x(n - i + 1)).\end{aligned}$$

Additionally, write $h(x) = \text{LO}(x) + \text{TZ}(x)$ and note that $0 \leq h(x) \leq n$. The payoff function for **Bilinear** is now given as follows.

$$f(x, y) = \begin{cases} \frac{1}{2} + h(x) - h(y) & \text{if } |x| = a \text{ and } |y| \neq b, \\ -\frac{1}{2} + h(x) - h(y) & \text{if } |x| \neq a \text{ and } |y| = b, \\ (|x| - a)(|y| - b) + h(x) - h(y) & \text{otherwise.} \end{cases}$$

The strict Nash equilibrium is given by $x^* = 1^a 0^{n-a}$ and $y^* = 1^b 0^{n-b}$. Note that this payoff function is isomorphic to the problem definitions appearing in [40, 31], except for the addition of the $\pm \frac{1}{2}$ (to ensure the Nash equilibrium is strict) and $h(x) - h(y)$ (to ensure that the Nash equilibrium is unique). All runs were executed using problem parameters $a = 35$ and $b = 10$.

C.2 PlantedBilinear

PlantedBilinear is a randomly instantiated zero-sum game. First, u and v are sampled uniformly at random from $\{0, 1\}^n$, a random $n \times n$ matrix A is instantiated by generating each entry uniformly at random from $[0, 1]$. The payoff function is then defined as

$$f(x, y) = \begin{cases} \varepsilon & \text{if } x = u \text{ and } y \neq v, \\ -\varepsilon & \text{if } x \neq u \text{ and } y = v, \\ 2x^\top Ay - x^\top Av - u^\top Ay & \text{otherwise.} \end{cases}$$

Where ε is chosen to satisfy $\varepsilon < |\{2x^\top Ay - x^\top Av - u^\top Ay : x \neq u \text{ and } y \neq v\}|$ (we used `min_float` from Ocaml's native `Float` module). The strict Nash equilibrium is given by $x^* = u$ and $y^* = v$. **PlantedBilinear** generalises **Bilinear**-type problem to forms more akin to that provided in [63]. All runs were executed using the same random seed to generate u , v , and A , so that all problem instances were identical.

C.3 MBJR_2024

Introduced in [47], we adopt the zero-sum formulation of [34] after projecting $\{0, 1\}^n$ onto $\{0, \dots, n\}$ using the function $|\cdot|$. Let $\Delta_{\min}, \Delta_1 \in \mathbb{R}$. The payoff function for MBJR_2024 is given as follows.

$$f(x, y) = \begin{cases} 0 & \text{if } |x| = n \text{ and } |y| = n, \\ 2\Delta_{\min} & \text{if } |x| = n \text{ and } |y| = n - 1, \\ -2\Delta_{\min} & \text{if } |x| = n - 1 \text{ and } |y| = n, \\ 2\Delta_1 & \text{if } |x| = n \text{ and } |y| < n - 1, \\ -2\Delta_1 & \text{if } |x| < n - 1 \text{ and } |y| = n, \\ 0 & \text{if } |x| = |y| < n, \\ 1 & \text{if } |y| < |x| < n, \\ -1 & \text{if } |x| < |y| < n. \end{cases}$$

The strict Nash equilibrium is given by $x^* = y^* = \mathbf{1}^n$. Our runs were executed using $\Delta_{\min} = 0.001$ and $\Delta_1 = 0.1$.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction claim the development of a theoretical technique for deriving upper and lower bounds on the threshold between stability and instability for population-based CoEAs with unary variation, as well as a number of applications of this tool to various algorithms and problem classes, comparison to empirical analysis, and a proven link to regret analysis. The promised tool is developed in Section 3.1 (with corresponding proofs in Appendices B.3 and B.4), the subsequent applications are given in Section 3.2 (with corresponding proofs throughout Appendix B), the empirical analysis is provided in Section 4, and the corresponding regret analysis is covered in Section 5 (with corresponding proof in Appendix B.8).

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The main limitations arise from the gaps between the upper and lower bounds on stability, our rigorous understand of to what extent such gaps are necessary, and the fact that only one class of solution concept (strict Nash equilibrium) is considered in detail. These are discussed in Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: All results have all assumptions stated precisely with respect to the definitions set out in Section 2. All proofs are complete and contained in Appendix B. Results upon which our proofs are dependent are stated and referenced in Appendix B.1.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: The algorithms examined empirically in Section 4 are simple and plainly stated in Algorithms 2 and 3 (alongside Algorithm 1). Methodological aspects including mutation operator, population size, number of runs, and the procedure followed to generate Figure 2 are all provided in Section 4 and complete descriptions of the problem instances can be found in Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

- (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
- (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
- (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: Python code that reproduces the empirical analysis in the paper is publicly available at <https://github.com/asbenford/stability-analysis-of-coeas>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Full specification of mutation operator, population size, number of runs, and statistical aspects of the procedure followed to generate Figure 2 are provided in Section 4

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: As all of the claims made in this paper are theoretical, the aim of the empirical analysis is to provide broad intuition about behaviour in circumstances where gaps between theoretical bounds make no specific prediction, rather than to scientifically test a claim or hypothesis. As the contributions of the paper were claims supported by rigorous proof rather than experimental data, statistical tests were not appropriate.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The relevant details for the computational resources used to produce Figure 2 are provided in Appendix C and signposted in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research has been conducted fully in accordance with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: In Section 1 we consider how theoretical guarantees can instill greater confidence in the output of CoEAs. We also explain how the quantitative nature of the main theoretical contribution makes it a valuable tool for practitioners applying such algorithms. No foreseeable negative societal impacts arise beyond anything applicable to all theoretical works advancing understanding of CoEAs.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: This paper does not use existing assets.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.