Runtime Bounds for a Coevolutionary Algorithm on Classes of Potential Games

Mario Hevia Fajardo
University of Birmingham
Birmingham, United Kingdom
mario_hevia@hotmail.com

Jamal Toutouh
University of Malaga
Malaga, Spain
jamal@uma.es

Erik Hemberg
MIT
Cambridge, USA
hembergerik@csail.mit.edu

Una-May O'Reilly
MIT
Cambridge, USA
unamay@csail.mit.edu

Per Kristian Lehre
University of Birmingham
Birmingham, United Kingdom
p.k.lehre@bham.ac.uk

Abstract

Coevolutionary algorithms are a family of black-box optimisation algorithms with many applications in game theory. We study a coevolutionary algorithm on an important class of games in game theory: potential games. In these games, a real-valued function defined over the entire strategy space encapsulates the strategic choices of all players collectively. We present the first theoretical analysis of a coevolutionary algorithm on potential games, showing a runtime guarantee that holds for all exact potential games, some weighted and ordinal potential games, and certain non-potential games. Using this result, we show a polynomial runtime on singleton congestion games. Furthermore, we show that there exist games for which coevolutionary algorithms find Nash equilibria exponentially faster than best or better response dynamics, and games for which coevolutionary algorithms find *better* Nash equilibria as well. Finally, we conduct experimental evaluations showing that our algorithm can outperform widely used algorithms, such as better response on random instances of singleton congestion games, as well as fictitious play, counterfactual regret minimisation (CFR), and external sampling CFR on dynamic routing games.

CCS Concepts

• **Theory of computation → Evolutionary algorithms**; **Adversary models**.

Keywords

coevolution, evolutionary algorithms, game theory, potential games

ACM Reference Format:
Mario Hevia Fajardo, Jamal Toutouh, Erik Hemberg, Una-May O'Reilly, and Per Kristian Lehre. 2025. Runtime Bounds for a Coevolutionary Algorithm on Classes of Potential Games. In *Foundations of Genetic Algorithms XVIII (FOGA '25), August 27–29, 2025, Leiden, Netherlands.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3729878.3746616

ACM ISBN 979-8-4007-1859-5/2025/08
https://doi.org/10.1145/3729878.3746616

1 Introduction

Potential games [28] are an important class of non-cooperative games in game theory characterised by a function (*potential function*) whose maximisers are Nash equilibria. These games can be classified into exact, weighted, and ordinal, depending on the relationship between the potential function and the players' preferences. Potential games model many important problems in road pricing [35], telecommunications [36], cyber-security [12], among others.

Potential games are significant in game theory in part because the existence of a potential function implies the existence of a (not necessarily unique) pure Nash equilibrium, and moreover, simple learning dynamics (such as best response) provably converge to a Nash equilibrium in such games. Another advantage of the existence of a potential function is that it may enable the application of optimisation algorithms. If the potential function is known, then mathematical tools may be used. Otherwise, a black-box optimisation algorithm such as coevolutionary algorithms can be employed.

Coevolutionary algorithms have been successfully implemented for a wide range of applications, such as software engineering [1], cyber-security [29] and robotics [33]. Additionally, simple coevolutionary algorithms have been provably shown to efficiently find Nash equilibria in some classes of zero-sum games with local [2] and global intransitive cycles [14, 15, 20].

Despite the clear connection between potential games and optimisation, rigorous analyses of coevolutionary algorithms on this class of games or any of its subclasses are, to the best of our knowledge, non-existent. However, such analyses are crucial for providing performance guarantees and offering more tools (and guidance) to practitioners who deal with these types of games.

In this work we use runtime analysis to provide performance guarantees for a coevolutionary algorithm on a broad class of games that includes all exact potential games, some weighted and ordinal potential games and certain non-potential games. We refer to this class as *level games*. We also investigate whether coevolutionary algorithms can find better Nash equilibria (Nash equilibria with higher potential value) as well as asymptotically faster than best response dynamics.

More specifically, we present a simple population-based coevolutionary algorithm with a novel binary relation on strategy profiles, along with a general method for obtaining upper bounds on the expected optimization time (i.e., the expected number of payoff evaluations required to find a Nash equilibrium or any target set) of the proposed algorithm for all level games. Furthermore, if a level game satisfies certain reasonable constraints, we provide a polynomial runtime guarantee. Using this approach, we derive novel polynomial bounds for singleton congestion games. Our results can easily be translated to other types of coevolutionary algorithms using the same binary relation on strategy profiles. Additionally, we provide an example potential game where the coevolutionary algorithm finds a Nash equilibrium exponentially faster than best or better response dynamics. Moreover, we present a game where the coevolutionary algorithm, with high probability, finds Nash equilibria with a higher potential value (when maximising the potential).

Finally, we conduct experimental evaluations on random instances of singleton congestion games and dynamic routing games. For singleton congestion games, the experiments indicate that, despite an initial slowdown, our coevolutionary algorithm finds either a similar Nash equilibrium for monotone cost functions or a statistically significantly better Nash equilibrium for non-monotone cost functions compared to better response dynamics that only use pure strategies. For dynamic routing games our experiments show that our coevolutionary algorithm finds Nash equilibrium statistically significantly faster than fictitious play, counterfactual regret minimisation (CFR), and external sampling CFR.

The paper is organised as follows. Section 2 discusses related work on potential games and runtime analysis of co-evolutionary algorithms. Section 3 formally defines potential games and the classes of games considered in the paper. The section also introduces the Non-cartesian Potential CoEA and the level-based theorem used for its analysis. Section 4 provides a general upper bound on the runtime of NCP-CoEA on *level games*, a broad class of games we introduce here, then applies this to the special case of singleton congestion games. We then show that NCP-CoEA has exponentially better runtime than the classical best response learning dynamics on games with long paths. Furthermore, on some game classes, we show that NCP-CoEA can escape local optima, leading to better Nash equilibria with higher potential value than best response dynamics. Section 5 complements the theoretical analysis with an experimental study. Finally, Section 6 concludes the paper.

2 Related Work

Potential games. Potential games (see Section 3 for a formal definition) cover a wide class of interesting games and have been used to tackle problems in engineering [13, 35, 36] and computer science [12]. Monderer and Shapley [28] showed that every potential game has at least one pure (non-randomised) Nash equilibrium. Additionally, Monderer and Shapley [28] showed that all improvement paths are finite for every potential game, where an improvement path is a sequence of moves in which one player changes its strategy towards a new strategy that improves their utility. This characteristic allows convergence properties for best or better response dynamics [28] and fictitious play [27] on these games and similar results for *near-potential games* [5], i.e., some non-potential games which are "close" to potential games. Nonetheless, convergence does not guarantee efficient optimisation, Fabrikant et al. [10] showed that potential games are PLS-complete and Panageas et al. [30] showed that fictitious play can take exponential time to reach a Nash equilibrium. A corollary of the proof in [10] is that there exist examples with exponentially long shortest paths where best and better response dynamics need exponential number of steps. We discuss this further in Section 4.2.

Due to their importance in game theory, potential games and their characteristics have been studied in many papers (e.g. [3, 5, 16, 34]); we refer the reader to the surveys [11, 26].

Runtime of coevolutionary algorithms. Coevolutionary algorithms have been used in several domains [1, 29, 33]. Nonetheless, there is a small (but increasing) number of rigorous runtime analyses. Coevolutionary algorithms can be broadly divided into cooperative and competitive coevolutionary algorithms. Cooperative coevolutionary algorithms use two or more populations to solve traditional optimisation problems, and are therefore not directly amenable to the game-theoretic setting we are interested in here.

Lehre [20] analysed for the first time the runtime of a *competitive* coevolutionary algorithm. He showed that the population-based coevolutionary algorithm called PDCoEA finds an ε-approximate Nash equilibrium of a Bilinear game in expected polynomial time. Similarly, Hevia Fajardo and Lehre [14] showed that the $(1, \lambda)$ CoEA finds a solution *close* to a Nash equilibrium on a Bilinear game.

Lehre and Lin [23] analysed a coevolutionary algorithm on a binary payoff game, showing that the $(1, \lambda)$ CoEA can efficiently find an ε approximation on a benchmark problem.

Hevia Fajardo et al. [15] and the follow up Lehre and Lin [22] showed that a randomised local search coevolutionary algorithm is able to find a Nash equilibrium for the Bilinear game; however, due to a lack of population, it forgets it quickly. In a similar fashion, for a class of games with local intransitive cycles, Benford and Lehre [2] proved that a broad class of coevolutionary algorithms that do not use populations need super-polynomial runtime, while a population-based coevolutionary algorithm finds a Nash equilibrium in polynomial time.

3 Preliminaries

We denote the first n natural numbers as the set $[n] := \{1, \ldots, n\}$. We denote the natural logarithm as $\ln(\cdot)$ and the binary logarithm as $\log(\cdot)$. We use asymptotic notation ("Big-O" notation) to provide bounds on the growth rate of the algorithm's runtime as a function of the number of players and strategies.

Potential games. In this work, we consider games in normal form represented as a tuple $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$ where $\boldsymbol{I} = \{1, 2, \ldots, I\}$ is the set of players where I is the number of players, $\boldsymbol{S} = S_1 \times S_2 \times \cdots \times S_I$ is the set of strategy profiles, with S_i representing the set of pure strategies available to player i for $i \in \boldsymbol{I}$ and $\boldsymbol{u} = \{u_1, u_2, \ldots, u_I\}$ is the set of payoff (or utility) functions, where $u_i : \boldsymbol{S} \to \mathbb{R}$ is the payoff function of player $i \in \boldsymbol{I}$.

We denote a strategy profile as $\sigma = (s_1, s_2, \ldots, s_I)$ such that $s_1 \in S_1, s_2 \in S_2, \ldots, s_I \in S_I$[1]. For convenience we often use the notation σ_{-i} to denote the tuple $(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_I)$. Given a strategy profile σ and a strategy s_i' we denote by $(\sigma_{-i}, s_i') \in \boldsymbol{S}$ the strategy profile $(s_1, \ldots, s_{i-1}, s_i', s_{i+1}, \ldots, s_I)$ and by a slight abuse of notation we can also write σ as (σ_{-i}, s_i).

[1]Since our focus is on potential games, which are guaranteed to have at least one pure Nash equilibrium, we do not consider *mixed* strategies in this work.

We say that a strategy profile $\sigma^* = (s_1^*, s_2^*, \ldots, s_I^*)$ is a Nash Equilibrium with respect to a game $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$ iff for every player $i \in \boldsymbol{I}$ it holds that $u_i(\sigma^*_{-i}, s_i^*) \geq u_i(\sigma^*_{-i}, s_i)$ for all possible strategies $s_i \in S_i$.

In this work we focus on potential games [28], though our results also extend to certain non-potential games. A game is called a potential game if there exists a real-valued function on the strategy space (called a *potential function*) that measures the difference in a player's payoff when they unilaterally deviate. Monderer and Shapley [28] divided potential games into several classes, here we show three of them.

Definition 3.1. Let $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$ be a normal form game. Then, $\boldsymbol{G}$ is

- an *exact potential game* if there exists a function $P : \boldsymbol{S} \to \mathbb{R}$ such that for every $i \in \boldsymbol{I}$, $u_i(\sigma_{-i}, s_i) - u_i(\sigma_{-i}, s_i') = P(\sigma_{-i}, s_i) - P(\sigma_{-i}, s_i')\ \forall s_i, s_i' \in S_i$.
- a *weighted potential game* if there exists a function $P : \boldsymbol{S} \to \mathbb{R}$ such that for every $i \in \boldsymbol{I}$ there exist $w_1, \ldots, w_I \in \mathbb{R}^+$ such that $\forall s_i, s_i' \in S_i$, $u_i(\sigma_{-i}, s_i) - u_i(\sigma_{-i}, s_i') = w_i[P(\sigma_{-i}, s_i) - P(\sigma_{-i}, s_i')]\ \forall s_i, s_i' \in S_i$.
- an *ordinal potential game* if there exists a function $P : \boldsymbol{S} \to \mathbb{R}$ such that for every $i \in \boldsymbol{I}$, $u_i(\sigma_{-i}, s_i) - u_i(\sigma_{-i}, s_i') > 0 \Leftrightarrow P(\sigma_{-i}, s_i) - P(\sigma_{-i}, s_i') > 0\ \forall s_i, s_i' \in S_i$.

Exact potential games are a subset of weighted potential games, which in turn are a subset of ordinal potential games.

In this work, we assume that the algorithms have *oracle access* to the game $\boldsymbol{G}$, that is, the algorithm can evaluate the payoff $u_i(\sigma)$ (payoff/function evaluations) of any player $i \in \boldsymbol{I}$ and for any strategy profile $\sigma \in \boldsymbol{S}$, however, it does not have access to any other information about $\boldsymbol{G}$ including the expressions for the payoff functions $\boldsymbol{u}$, the potential function P, or their derivatives.

We use this assumption because finding the potential function of a game is in itself a hard optimisation problem [5, 6, 16] and if the potential function or the payoff functions are known it might be more efficient to use mathematical tools.

To design our coevolutionary algorithm we rely on the following binary relation on strategy profiles.

Definition 3.2. Given a game $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$ and two strategy profiles $\sigma = (s_1, s_2, \ldots, s_I)$ and $\sigma' = (s_1', s_2', \ldots, s_I')$ with $\sigma, \sigma' \in \boldsymbol{S}$, we say that $\sigma \triangleright_G \sigma'$, iff

$$0 < \sum_{i=1}^{I} u_i(s_1', \ldots, s_{i-1}', s_i, \ldots, s_I) - u_i(s_1', \ldots, s_i', s_{i+1}, \ldots, s_I).$$

If $\geq$ is used instead of $>$, we denote this as $\sigma \trianglerighteq_G \sigma'$.

This relation is useful for two reasons. Firstly, it enables the characterisation of Nash equilibria, as is shown in the following lemma.

Lemma 3.3. *Let $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$ be a game. A strategy $\sigma^* = (s_1^*, s_2^*, \ldots, s_I^*)$ is a Nash Equilibrium with respect to a game $\boldsymbol{G}$ iff for every player $i \in \boldsymbol{I}$ it holds that $(\sigma^*_{-i}, s_i^*) \trianglerighteq_G (\sigma^*_{-i}, s_i)$ for all possible strategies $s_i \in S_i$.*

Proof. By the definition of the binary relation, $(\sigma^*_{-i}, s_i^*) \trianglerighteq_G (\sigma^*_{-i}, s_i)$ implies that

$$u_i(s_1^*, \ldots, s_I^*) \quad - \quad u_i(s_1^* \ldots s_{i-1}^*, s_i, s_{i+1}^*, \ldots, s_I^*) \quad \geq \quad 0, \quad (1)$$

for every player $i \in \boldsymbol{I}$ and every strategy $s_i \in S_i$. This condition corresponds to the definition of a Nash Equilibrium. □

Secondly, for any exact potential game $\boldsymbol{G}$ with potential function P the binary relation $\sigma \triangleright_G \sigma'$ implies $P(\sigma) \geq P(\sigma')$, this is shown in the following lemma by using the fact that for all players $i \in \boldsymbol{I}$, $u_i(\sigma_{-i}, s_i) - u_i(\sigma_{-i}, s_i') = P(\sigma_{-i}, s_i) - P(\sigma_{-i}, s_i')\ \forall s_i, s_i' \in S_i$.

Lemma 3.4. *Let $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$ be an exact potential game with potential function P. Given two strategy profiles $\sigma = (s_1, s_2, \ldots, s_I)$ and $\sigma' = (s_1', s_2', \ldots, s_I')$ with $\sigma, \sigma' \in \boldsymbol{S}$,*

$$\sigma \triangleright_G \sigma' \qquad \Leftrightarrow \qquad P(\sigma) > P(\sigma'). \quad (2)$$

Proof. By the definition of exact potential games, for all players $i \in \boldsymbol{I}$, $u_i(\sigma_{-i}, s_i) - u_i(\sigma_{-i}, s_i') = P(\sigma_{-i}, s_i) - P(\sigma_{-i}, s_i')\ \forall s_i, s_i' \in S_i$. Hence, for all $i \in \boldsymbol{I}$

$$\begin{aligned} &u_i(s_1', \ldots, s_{i-1}', s_i, \ldots, s_I) - u_i(s_1' \ldots, s_i', s_{i+1}, \ldots, s_I) \\ \equiv &P(s_1', \ldots, s_{i-1}', s_i, \ldots, s_I) - P(s_1' \ldots, s_i', s_{i+1}, \ldots, s_I) \end{aligned}$$

and

$$\begin{aligned} \sigma \triangleright_G \sigma' \equiv &P(s_1, s_2, \ldots, s_I) - P(s_1', s_2, \ldots, s_I) \\ &+P(s_1', s_2, \ldots, s_I) - P(s_1', s_2', s_3, \ldots, s_I) + \ldots \\ &+P(s_1', \ldots, s_{i-1}', s_i, \ldots, s_I) \\ &-P(s_1' \ldots, s_i', s_{i+1}, \ldots, s_I) + \ldots \\ &+P(s_1', \ldots, s_{I-1}', s_I) - P(s_1', s_2', \ldots, s_I') > 0. \end{aligned}$$

We note that the intermediate summands cancel out, hence, $\sigma \triangleright_G \sigma' \equiv P(\sigma) > P(\sigma')$. □

Although this relation aligns with the potential function on all exact potential games, it does not work on all potential games. In the following we show an example weighted potential game where there exist two strategy profiles $\sigma, \sigma' \in \boldsymbol{S}$ such that $P(\sigma') > P(\sigma)$ and $\sigma \triangleright_G \sigma'$.

Example 3.5. Let $P(s_1, s_2) = 1 + 3s_1 + 1.5s_2 - 2.5s_1s_2$, and $w_1 = 0.1$, $w_2 = 1$. Then, $\boldsymbol{G} = (\boldsymbol{I} = \{1, 2\}, \boldsymbol{S} = \{\{0, 1\} \times \{0, 1\}\}, \boldsymbol{u} = \{0.1(1 + 3s_1 + 1.5s_2 - 2.5s_1s_2), 1 + 3s_1 + 1.5s_2 - 2.5s_1s_2\})$ is a weighted potential game with potential function P and weights w_1, w_2. Let $\sigma = (0, 0)$ and $\sigma' = (1, 1)$ then, $P(\sigma') > P(\sigma)$ and $\sigma \triangleright_G \sigma'$.

For this game, we can see that $u_1(s_1, s_2) - u_1(s_1', s_2) + u_2(s_1', s_2) - u_2(s_1', s_2') = 0.1(1) - 0.1(4) + 1(4) - 1(3) = 0.7$, hence $\sigma \triangleright_G \sigma'$, but $3 = P(1, 1) = P(\sigma') > P(\sigma) = P(0, 0) = 1$.

Our analysis will be applicable to all games for which Lemma 3.4 holds. While this includes all exact potential games, it also holds true for some weighted and ordinal potential games[2]. Additionally, we believe that it is still useful for certain non-potential games such as near-exact potential games [5] without the need to find the closest exact potential game.

Algorithm. In this work we propose a population-based coevolutionary algorithm (similar to PDCoEA [20]) that uses the binary relation $\triangleright_G$ from Definition 3.2 (Algorithm 1). The algorithm starts

[2] Examples are shown in Appendix A.

with a population P_0 of λ strategy profiles (individuals) sampled u. a. r. from the set of all possible strategy profiles $\boldsymbol{S}$. Then each iteration t it generates a new population P_{t+1} based on the current one P_t. Each individual in P_{t+1} is generated independently using a binary tournament selection, that is, by selecting the best (with respect to $\triangleright_G$) of two individuals sampled uniformly at random (u. a. r.) from P_t and mutating it using a variation operator. Algorithm 1 accepts any variation operator $\mathrm{mut}(\cdot) : \boldsymbol{S} \to \boldsymbol{S}$, that is an operator that receives a strategy $\sigma \in \boldsymbol{S}$ and modifies it into a new strategy profile $\sigma' \in \boldsymbol{S}$. For our theoretical and experimental analyses the mutation operator changes the action of a player $i \in \boldsymbol{I}$ independently with probability χ/I. We assume that the actions do not have any particular topology, therefore the new action of player i is selected u. a. r. from the allowed actions S_i. A final solution can be extracted by selecting the most common solution in the population. This approach is justified by the level-based theorem (cf. Section 3.1), which ensures that if a Nash equilibrium is found, it will occupy at least a constant fraction of the population, and a similar trend applies to the best solution before reaching a Nash equilibrium.

We use a population-based coevolutionary algorithm because it has been argued [8], and both empirically [21] and theoretically [2] shown that populations can improve performance when optimising complex games. Additionally, we use tournament selection because it is easy to implement, works well on parallel architectures, and has been rigorously shown to be able to deal with noisy environments [25] and efficiently escape local optima [9] on similar algorithms.

Algorithm 1 Non-cartesian Potential CoEA (NCP-CoEA)

Require: Exact potential game $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$.
Require: Population size $\lambda \in \mathbb{N}$.
Require: Mutation operator $\mathrm{mut}(\cdot) : \boldsymbol{S} \to \boldsymbol{S}$.
  for $i \in [\lambda]$ **do** Sample $P_0(i) \sim \mathrm{Unif}\ (\boldsymbol{S})$
  for $t \in \mathbb{N}$ until termination criterion met **do**
    for $j \in [\lambda]$ **do**
      Sample $\sigma_1, \sigma_2 \sim \mathrm{Unif}\ (P_t)$
      if $\sigma_1 \triangleright_G \sigma_2$ **then** $\sigma := \sigma_1$ **else** $\sigma := \sigma_2$
      Set $P_{t+1}(j) := \mathrm{mut}(\sigma)$
    end for
  end for

For completeness, we also give the pseudo-code of best or better response dynamics in Algorithm 2. The algorithm starts with an initial strategy profile where each player has chosen some strategy. Then it repeats the following steps until the termination criterion is met. The algorithm selects a player according to a selection rule, after which it finds the best or a better response strategy for this player to the current strategies of its opponents and substitutes this strategy into the current strategy profile.

3.1 Level-based Analysis

The level-based theorem [7] is a general technique for obtaining upper bounds on the expected runtime of a stochastic process of the form of Algorithm 3. Algorithm 3 describes a population-based stochastic process $(P_t)_{t\in\mathbb{N}}$ where for each iteration $t \in \mathbb{N}$,

Algorithm 2 Best or Better Response Dynamics

Require: Game $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$.
Require: Initial strategy profile $\sigma_0 \in \boldsymbol{S}$.
Require: Player selection rule (e.g., round-robin, random).
  for $t \in \mathbb{N}$ until termination criterion met **do**
    Select player $i \in \boldsymbol{I}$ according to selection rule
    if response rule = "best" **then**
      Compute best response: $s_i^* \in \arg\max_{s_i \in S_i} u_i(s_i, \sigma_{t,-i})$
    else if response rule = "better" **then**
      Select $s_i^* \in \boldsymbol{S}_i$ such that $u_i(s_i^*, \sigma_{t,-i}) > u_i(\sigma_{t,i}, \sigma_{t,-i})$
    end if
    $\sigma_t := (s_i^*, s_{-i})$
  end for

$P_t = (\sigma_1, \ldots, \sigma_\lambda) \in \boldsymbol{S}^\lambda$ is a vector of λ individuals (strategy profiles in this work). The main assumption is that P_{t+1} is sampled from some distribution $D(P_t)$. The distribution D describes the randomised process that dictates the generation of new individuals. Algorithm 3 covers many search heuristics including Algorithm 1 and has been used to analyse many evolutionary algorithms on various optimisation problems [7, 9, 24, 25].

Algorithm 3 Population-Based Process

Require: A finite state space $\boldsymbol{S}$, a population size $\lambda \in \mathbb{N}$ and initial population $P_0 \in \boldsymbol{S}^\lambda$.
  for $t \in \mathbb{N}$ until termination criterion met **do**
    Sample $P_{t+1}(i) \sim D(P_t)$ independently $\forall i \in [\lambda]$
  end for

To apply the level-based theorem the search space $\boldsymbol{S}$ must be partitioned into m subsets $(A_1, \ldots, A_m)$ that are called levels, where the last level A_m contains only the target set. If certain conditions are met, the level-based theorem gives an upper bound on the expected number of sampled individuals until the algorithm discovers an element in A_m.

Theorem 3.6 (Level-based theorem [7]). *Given a partition $(A_1, \ldots, A_m)$ of $\boldsymbol{S}$, define $T := \min\{t\lambda \mid |P_t \cap A_m| > 0\}$, where for all $t \in \mathbb{N}$, $P_t \in \boldsymbol{S}^\lambda$ is the population in generation t. If there exist $z_1, \ldots, z_{m-1}, \delta \in (0, 1]$, and $\gamma_0 \in (0, 1)$ such that for any population $P \in \boldsymbol{S}^\lambda$*

(1) for each level $j \in [m-1]$, if $|P_t \cap A_{\geq j}| \geq \gamma_0\lambda$, then $\Pr_{\sigma' \sim D(P)}[\sigma' \in A_{\geq j+1}] \geq z_j$
(2) for each level $j \in [m-2]$, and all $\gamma \in (0, \gamma_0]$ if $|P_t \cap A_{\geq j}| \geq \gamma_0\lambda$ and $|P_t \cap A_{\geq j+1}| \geq \gamma\lambda$, then $\Pr_{\sigma' \sim D(P)}[\sigma' \in A_{\geq j+1}] \geq (1+\delta)\gamma$
(3) and the population size $\lambda \in \mathbb{N}$ satisfies $\lambda \geq \left(\frac{4}{\gamma_0\delta^2}\right) \ln\left(\frac{128m}{z_*\delta^2}\right)$, where $z_* := \min_{j\in[m-1]}\{z_j\}$

then $\mathrm{E}[T] \leq \left(\frac{8}{\delta^2}\right) \sum_{j=1}^{m-1} \left(\lambda \ln\left(\frac{6\delta\lambda}{4+z_j\delta\lambda}\right) + \frac{1}{z_j}\right)$.

Informally, the conditions of Theorem 3.6 require that:

(1) Condition (G1): there must be a non-zero probability of sampling an individual above the "current level" j.

(2) Condition (G2): if a fraction of the population is above the "current level", the probability of sampling an individual above this level is larger than that fraction.
(3) Condition (G3): the population size is sufficiently large.

Conditions (G1) and (G2) ensure that the process advances through the levels until the target set is reached, while Condition (G3) helps prevent fallbacks.

4 Runtime Analysis

In this section we first introduce level games. This class of games are all games in which the strategy space $\boldsymbol{S}$ can be partitioned into ordered non-empty subsets (levels) $A_1, \ldots, A_m$, where A_m consists solely of pure Nash equilibria, though not all Nash equilibria are necessarily included. These subsets are ordered with respect to the binary comparison $\triangleright_G$, that is, for every $1 \le j < m$, $\forall \sigma \in A_{j+1}$ and $\forall \sigma' \in A_j$ we have $\sigma \triangleright_G \sigma'$ independent of the order of the players. We note that by definition all level games have at least one pure Nash equilibrium. Furthermore, by Lemma 3.4 this class of games includes all exact potential games, as the search space can be partitioned by the potential value. Additionally, it includes some weighted and ordinal potential games and non-potential games. We give examples of level games in each of these classes in Appendix A.

We show a bound for the expected number of payoff evaluations Algorithm 1 needs to find a Nash equilibrium (or any other target set) that holds if there is a partition of the search space such that there is a non-zero probability of mutating a strategy profile $\sigma \in A_j$ into a strategy profile $\sigma' \in A_{\ge j+1}$ for all levels $j \in [1, m-1]$ and the probability of mutating a strategy profile $\sigma \in A_j$ into $\sigma' \in A_{\ge j}$ is a sufficiently large value p_0. In the following we formalise these assumptions.

Assumption 1. *For all strategy profiles $\sigma \in A_{\ge j}$ and all $1 \le j < m$ the probability that a new strategy profile $\sigma' \in A_{\ge j+1}$ is created by* $\mathrm{mut}(\sigma)$ *is at least $h_j > 0$.*

Assumption 2. *There exist a constant $\delta > 0$ such that for all strategy profiles $\sigma \in A_j$ and all $1 \le j < m$ the probability that a new strategy profile $\sigma' \in A_{\ge j}$ is created by* $\mathrm{mut}(\sigma)$ *is at least $p_0 = 1/2 + \delta$.*

Assumption 3. *The population size $\lambda = O(\log(m))$ and is at least $\frac{32p_0}{(2p_0-1)^3} \ln\left(\frac{128m}{(1-1/(2p_0))^2 \min_{j\in[m-1]}(h_j)}\right)$.*

These assumptions are related to the conditions of Theorem 3.6. It is easy to see that Assumption 1 fulfils Condition 1 with $z_j := \gamma_0^2 h_j$. For Condition 2 we need to argue a bit further. This is done in the following lemma.

Lemma 4.1. *Let $\delta > 0$ be the constant from Assumption 2, and $0 < \gamma_0 \le \frac{2\delta}{2\delta+1}$. Under Assumption 2, for each level $j \in [m-2]$, and all $\gamma \in (0, \gamma_0]$ if $|P \cap A_j| \ge \gamma_0\lambda$ and $|P \cap A_{j+1}| \ge \gamma\lambda$, then there exists a constant $\delta > 0$ such that* $\Pr[\sigma' \in A_{j+1}] \ge (1+\delta)\gamma$

Proof. By Assumption 2 the probability of mutating $\sigma \in A_{j+1}$ into $\sigma' \in A_{j+1}$ is at least p_0. Since there are at least $\gamma\lambda$ strategy profiles in the population within level A_{j+1}, we have $\Pr[\sigma' \in A_{j+1}] \ge (1-(1-\gamma)^2)p_0 = \gamma(2-\gamma)p_0$. By Assumption 2 and $0 < \gamma \le \gamma_0 \le \frac{2\delta}{2\delta+1}$ we obtain

$$\Pr\left[\sigma' \in A_{j+1}\right] \ge \gamma\left(2 - \frac{2\delta}{2\delta+1}\right)(1/2+\delta)$$
$$= \gamma\left(1 + 2\delta - \frac{2\delta^2+\delta}{2\delta+1}\right) = \gamma(1+\delta). \qquad \square$$

Finally, using $\gamma_0 = 2\delta/(2\delta+1)$ from Lemma 4.1, and the other two assumptions we can see that Assumption 3 implies Condition 3.

With Lemma 4.1 and the assumptions we use Theorem 3.6 to show the main theorem of this work.

Theorem 4.2. *Let $\delta = p_0 - 1/2 > 0$ be a constant, $\gamma_0 = 1 - 1/(2p_0)$ and $z_j = \gamma_0^2 h_j$ for all $1 \le j < m$ where h_j and p_0 comes from Assumptions 1 and 2. Let $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$ be a level game. Given a partition $(A_1, \ldots, A_m)$ of $\boldsymbol{S}$, define $T := \min\{2It\lambda \mid |P_t \cap A_m| > 0\}$, where for all $t \in \mathbb{N}$, $P_t \in \boldsymbol{S}^\lambda$ is the population of Algorithm 1 in generation t. Under Assumptions 1, 2 and 3 the* $\mathrm{E}[T]$ *of Algorithm 1 on $\boldsymbol{G}$ is at most*

$$\left(\frac{16I}{\delta^2}\right)\sum_{j=1}^{m-1}\left(\lambda \ln\left(\frac{6\delta\lambda}{4+z_j\delta\lambda}\right) + \frac{1}{z_j}\right)$$
$$= O\left(mI\lambda\ln\lambda + \frac{mI}{\min_{j\in[m-1]} z_j}\right).$$

Proof. We aim to use the Level-based Theorem (Theorem 3.6). By Assumptions 1 and 3 the conditions 1 and 3 are met. By Lemma 4.1 and Assumption 2, condition 2 is met. Hence, by applying Theorem 3.6 and noting that each time we sample a new strategy the algorithm uses at most $2I$ payoff evaluations we obtain the claimed runtime. $\square$

We note that Theorem 4.2 gives a runtime bound for Algorithm 1 on all exact potential games, some weighted and ordinal potential games and certain non-potential games. Moreover, the conditions for polynomial runtime are reasonable and analogous to what best or better response dynamics need to achieve a polynomial runtime: a convergence path (number of levels) of polynomial length where it is *easy* to progress.

4.1 Polynomial Runtime on Singleton Congestion Games

In this section we showcase a use case for Theorem 4.2 by analysing the runtime of Algorithm 1 on a class of games named singleton congestion games (SCG).

A *congestion game* is an I-player game where each player's strategy is a selection (subset) of resources from the set of available resources R with $|R| = k$. Each resource r have a cost function $c_r : \{1, 2, \ldots, I\} \to \mathbb{R}$ that only depends on the number of players using that resource that we will refer as v_r, and the cost of a strategy is the sum of the costs of the selected resources by the player. We will assume that players are attempting to minimise their costs. In normal form a congestion game is a game $\boldsymbol{G} := (\boldsymbol{I}, \boldsymbol{S}, \boldsymbol{u})$, where for each player i, $S_i \subseteq 2^R$ and $u_i = -\sum_{r\in R} c_r(v_r)$.

Rosenthal [31] showed that congestion games are exact potential games, with a potential function $P = \sum_{r\in R}\sum_{\ell=1}^{v_r} c_r(\ell)$. Monderer and Shapley [28] proved the converse: every exact potential game is a congestion game. Hence, we can apply Theorem 4.2 to this class of games and its sub-classes. Given that Fabrikant et al. [10]

showed that congestion games are intractable, one cannot expect to find a polynomial runtime for all congestion games. Therefore, we work with a tractable subclass.

Singleton congestion games are games where each player must use exactly one resource. In these games the cost functions can be represented as a cost matrix C with entries $C(r, j) = (c_r(j))_{rj}$. The rows in the matrix represent the resource $r \in R$ and the columns the number of players using that resource. It is clear that the matrix has Ik entries.

Similar to Ieong et al. [18], for every entry in the cost matrix C we define its rank as follows.

Definition 4.3 ([18]). Given a cost matrix C, the rank of cost entry $C(r, j)$ is ℓ if there are $\ell - 1$ distinct cost values (not entries) in C that are less than $C(r, j)$

With this definition and following the proof ideas of [18] we can show that Algorithm 1 finds a Nash equilibrium in polynomial time for reasonable parameters.

Theorem 4.4. *Let **G** be a singleton congestion game with $I \geq 2$ players and k resources. Consider Algorithm 1 with a population size $350 \ln\left(\frac{1568 I^3 k^2}{\chi e^{-\chi}}\right) \leq \lambda = O(\ln(Ik))$ and the mutation operator described in Section 3 with a constant $0 < \chi \leq 0.336288$. Then, Algorithm 1 reaches a Nash equilibrium in $O(I^4 k^2)$ expected payoff evaluations.*

The main proof idea is to show that there are at most $I^2 k$ different potential values. Then, we can partition the search space into $I^2 k$ levels. Assuming that the algorithm is at level j and the algorithm is not in a Nash equilibrium then the probability of mutating into a higher level is $\Omega(1/(Ik))$ and for a sufficiently small χ the probability of a mutation not changing the current profile strategy (and staying on the same level) is larger than $1/2$. This shows all assumptions needed to use Theorem 4.2.

Proof. Following [18] we transform the cost matrix C of the game by replacing each cost entry by its rank. We note that Algorithm 1 operates solely based on comparing the potential of a profile strategy with that of another, rather than considering the actual costs incurred by players. Therefore, the algorithm's behaviour remains consistent when applied to the game using the transformed cost matrix. The potential function P' resulting from the game with the transformed cost matrix can only have integral values between 1 and $I^2 k$ since the highest rank in the cost matrix is at most Ik and there are I players.

Then, we can define the levels for Theorem 4.2 as follows. All strategy profiles σ in level A_i have a potential $P'(\sigma) \leq m - i$ with $m \leq I^2 k$. Hence there are m levels and level A_m includes only strategy profiles with the minimum potential value.

If the algorithm is not in a Nash Equilibrium then there is at least one player that can unilaterally choose at least one strategy that is better than the current one. This strategy by definition would reduce the potential value and hence move towards a higher level. Therefore, it is sufficient for the mutation operator to modify only this player into the better strategy. This has a probability

$$\left(1 - \frac{\chi}{I}\right)^{I-1} \left(\frac{\chi}{I}\right) \left(\frac{1}{k}\right) \geq \frac{\chi e^{-\chi}}{Ik}.$$

Hence, Assumption 1 is fulfilled with $h_j := \frac{\chi e^{-\chi}}{Ik}$ for all $1 \leq j < m$.

To fulfil Assumption 2 we compute the probability that the mutation operator does not change any player, that is,

$$\left(1 - \frac{\chi}{I}\right)^{I} \geq \left(1 - \frac{\chi}{2}\right)^{2}.$$

To provide exact constants for Theorem 4.4, we aim to show $p_0 \geq 0.7$, but the proof still holds by replacing that constant with any other constant $p_0 > 1/2$. Choosing $0 < \chi \leq 0.336288$ fulfils Assumption 2 with $\delta = 0.2$.

Finally, Assumption 3 asks for $\lambda = O(\log(m))$ and

$$\lambda \geq \frac{32 p_0}{(2p_0 - 1)^3} \ln\left(\frac{128 m}{(1 - 1/(2p_0))^2 \min\{h_j\}}\right).$$

Plugging p_0 and h_j we obtain $\lambda = 350 \ln\left(\frac{1568 I^3 k^2}{\chi e^{-\chi}}\right)$, which by assumption of the statement is true.

Now we can apply Theorem 4.2 to obtain

$$\begin{aligned}
&\left(\frac{16I}{\delta^2}\right) \sum_{j=1}^{I^2 k} \left(\lambda \ln\left(\frac{6\delta\lambda I^2 k e^{\chi}}{4I^2 k e^{\chi} + \delta \gamma_0^2 \lambda \chi}\right) + \frac{Ike^{\chi}}{\gamma_0^2 \chi}\right) \\
&400I \sum_{j=1}^{I^2 k} \left(\lambda \ln\left(\frac{58.8\lambda I^2 k e^{\chi}}{196 I^2 k e^{\chi} + 0.8\lambda\chi}\right) + \frac{49Ike^{\chi}}{4\chi}\right) \\
&= O(I^4 k^2).
\end{aligned}$$

□

4.2 Long Path Games

As mentioned before, a corollary of the work of Fabrikant et al. [10] is that there exist potential games with exponentially long shortest improvement paths. That is, every sequence of steps where *exactly* one player changes its strategy towards a new strategy that improves its utility is exponentially long. We note that if you allow several players to change their strategies simultaneously then there could be "shortcuts" that reduce the size of the paths significantly, allowing to find Nash equilibria more efficiently. A similar result was shown by Rudolph [32] for evolutionary algorithms on exponentially long path optimisation functions where local mutation (changing only one bit) has exponential runtime, but global mutation (allowing more than one bit to be changed) has polynomial expected runtime.

Here we show an example game where this is true, more specifically, we will show a game that we call Path where best or better response dynamics with random initialisation and Bernoulli random revision sequences (at each step the next player is chosen at random, each with probability $1/I$) needs in expectation an exponential number of steps to reach a Nash equilibrium, but Algorithm 1 finds a Nash equilibrium in polynomial expected time. Despite being a constructed game, its analysis offers valuable insights into the behaviour of coevolutionary algorithms and the scenarios where their application can yield significant advantages.

The game Path has I players and each player has only two actions (0 or 1). Additionally, all players share the same utility function which will be the potential function.

As potential function we use a pseudo-Boolean function created by Horn et al. [17] and we consider each bit position a player. The function has a long path such that any two strategy profiles

σ_1, σ_2 adjacent on the path have a Hamming distance of exactly 1 (σ_1 differs from σ_2 by only one strategy) and any other point on the path have a Hamming distance of at least 2. The value of the potential function depends on the position of the strategy profile on the path.

The path is constructed recursively as follows. Given a base path with one player $\phi_1 = \{0, 1\}$ with starting point 0 and end point 1 we create a path for three players ϕ_3 by creating two copies C_{00} and C_{11} of ϕ_1 and for each point in the copies we prepend 00 in C_{00} and 11 in C_{11}. This creates two paths $C_{00} = \{000, 001\}$ and $C_{11} = \{110, 111\}$. Then we add a bridge point B_{01} by grabing the last point of ϕ_1 and prepending 01. Finally ϕ_3 is constructed by concatenating C_{00}, B_{01} and reversed C_{11}. Then $\phi_3 = \{000, 001, 011, 111, 110\}$.

Following the same steps, we can create a path ϕ_i from ϕ_{i-2} as long as i is odd. If the number of players is even we can ignore one of the players in the construction of the path. Given that every two players the path doubles in size, the path length is exponential in the number of players. Horn et al. [17] showed that the path length is $3 \cdot 2^{\lfloor (I-1)/2 \rfloor} - 1 = \Theta(2^{I/2})$.

Finally, following [17] for strategy profiles that are not in the path the potential function is equal to minus the number of 1 strategies, that we denote as $|\sigma|_1$ (we also denote $|\sigma|_0$ as the number of 0 strategies). This creates a "gradient" towards the start of the path, and ensures that for all strategy profiles there is always a player that can unilaterally change their strategy to improve their utility unless the strategy profile is the end point of the path. Then the potential function is defined as:

$$P(\sigma) := \begin{cases} \ell, & \text{if } \sigma \text{ is the } \ell\text{-th strat. profile of the path} \\ -|\sigma|_1, & \text{otherwise,} \end{cases}$$

and has its maximum value in the last strategy profile of the path, which is the Nash equilibrium.

In the following theorem, we show that best or better response dynamics is inefficient in this game.

Theorem 4.5. *Best or better response dynamics with random initialisation and Bernoulli random revision sequences needs in expectation $\Omega(2^{I/2})$ steps to find the Nash equilibrium of* Path.

Proof. The main idea of the proof is to show that the first time that the algorithm reaches the path it finds a point of C_{00} or B_{01}, that is, the first half of the path. Afterwards it must follow the path needing $\Omega(2^{I/2})$ steps.

Given that C_{00} and C_{11} are symmetric, if the first sampled strategy profile is on the path the probability of being in C_{00} is equal to C_{11}. Therefore with probability at least 1/2 the strategy profile is in the first half of the path ($C_{00} \cup B_{01}$).

It the first sampled strategy profile is not on the path, then every strategy profile has at least one player that can unilaterally improve its utility. By doing so the next strategy profile has either more 0 strategies or the path is reached. Again, from the symmetry of the path from the initial strategy profile the expected distance to either C_{00} or C_{11} is the same. Additionally, the first two players in the C_{11} always use strategy 1 but outside of the path the strategy 0 is preferred by all players. Therefore, with probability at least 1/2 the first strategy profile found by the best or better response dynamics is on the first half of the path and it needs to follow the path needing $\Omega(2^{I/2})$ steps. □

The proof of Theorem 4.5 relies on the fact that with constant probability the first strategy profile that the algorithm finds in the path is in the first half of the path and then it needs an exponential number of steps to find the unique Nash equilibrium.

In contrast Theorem 4.6 shows that Algorithm 1 needs a polynomial number of payoff evaluations in expectation to find the Nash equilibrium of Path. For simplicity, if the mutation operator in Algorithm 1 changes the action of a player it always selects the action that is not currently used.

Theorem 4.6. *Let $I \geq 2$ be the number of players. Algorithm 1 with a mutation probability $0 < \chi \leq 0.3336288$ and a population size $350 \ln\left(\frac{3136 I^2 (I+1)}{3\chi^2(1-\chi)}\right) \leq \lambda = O(\ln(I))$ needs $O(I^4)$ payoff evaluations in expectation to find the Nash equilibrium of* Path.

Proof. The first thing that we need to do is define the levels to use Theorem 4.2. Level A_j with $j \in [1, I]$ is composed of all strategy profiles that are not in the path and have $|\sigma|_0 = j - 1$, Afterwards, let O be the set of all strategy profiles in the path, then for $j \in [I+1, I+1+(I-1)/2]$ the levels are defined as:

$$\begin{aligned}
&A_{I+1} := \{00 * \cdots * \in O\} \cup \{01 * \cdots * \in O\} \\
&A_{I+2} := \{1111 * \cdots * \in O\} \cup \{1101 * \cdots * \in O\} \\
&A_{I+3} := \{110011 * \cdots * \in O\} \cup \{110001 * \cdots * \in O\} \\
&A_{I+4} := \{11000011 * \cdots * \in O\} \cup \{11000001 * \cdots * \in O\} \\
&A_{I+2+(I-5)/2} := \{1100\ldots0011* \in O\} \cup \{1100\ldots0001* \in O\} \\
&A_{I+2+(I-3)/2} := \{1100\ldots00001\} \\
&A_{I+2+(I-1)/2} := \{1100\ldots00000\}
\end{aligned}$$

Then the number of levels $m = I + 2 + (I-1)/2 = 3/2 \cdot (I+1)$.

Now we show that these levels fulfil Assumptions 1, 2 and 3. For each strategy profile in level A_j with $j \in [1, I]$ there is at least one strategy profile in level A_{j+1} that differs in only one strategy (changing one 1 strategy to 0). Therefore, it is sufficient to change one of the 1s strategies and do not change any other strategy. This has a probability

$$\left(\frac{\chi(I-j+1)}{I}\right)\left(1-\frac{\chi}{I}\right)^{I-1} \geq \left(\frac{\chi}{I}\right)\left(1-\frac{\chi}{I}\right)^{I-1} \geq \left(\frac{\chi}{I}\right)^2 (1-\chi).$$

Similarly, for each strategy profile in level A_j with $j \in [I, 3/2\cdot(I+1)]$ there is at least one strategy profile in level A_{j+1} that differs in exactly two strategies. Therefore, it is sufficient to change these two strategies and do not change any other strategy. This has a probability at least

$$\left(\frac{\chi}{I}\right)^2\left(1-\frac{\chi}{I}\right)^{I-2} \geq \left(\frac{\chi}{I}\right)^2\left(1-\frac{\chi}{I}\right)^{I} \geq \left(\frac{\chi}{I}\right)^2 (1-\chi).$$

Then, for all $1 \leq j < m$ Assumption 1 is met with

$$h_j = (\chi/I)^2 (1-\chi).$$

Now we will prove Assumption 2. It is clear that in order for a strategy profile $\sigma \in A_j$ to be mutated into a strategy profile $\sigma' \in A_j$

is sufficient that the mutation operator does not change any of the player's strategies. This has a probability

$$\left(1-\frac{\chi}{I}\right)^{I} \geq \left(1-\frac{\chi}{2}\right)^{2}.$$

Therefore, for $0 < \chi \leq 0.3336288$ fulfils Assumption 2 with $\delta = 0.2$.

Finally, Assumption 3 asks for $\lambda = O(\log(m))$ and

$$\lambda \geq \frac{32p_0}{(2p_0-1)^3}\ln\left(\frac{128m}{(1-1/(2p_0))^2\min\{h_j\}}\right).$$

Plugging p_0 and h_j we obtain $\lambda = 350\ln\left(\frac{3136I^2(I+1)}{3\chi^2(1-\chi)}\right)$, which by assumption of the statement is true.

Hence, we can apply Theorem 4.2 to obtain a bound on the expected number of payoff evaluations of at most

$$400I\sum_{j=1}^{3/2\cdot(I+1)-1}\left(\lambda\ln\left(\frac{58.8I^2\lambda}{196I^2+0.8\chi^2(1-\chi)\lambda}\right)+\frac{49I^2}{4\chi^2(1-\chi)}\right)$$
$$= O(I^4).$$

□

The proof of Theorem 4.6 relies on that every strategy profile in the path has a "parallel" strategy profile that is at a distance two (different strategies for exactly two players) from a strategy profile that is at least halfway in the path. Therefore, the algorithm is able to halve the distance of the path to the Nash equilibrium by *jumping* to this nearby strategy profiles. Then it just needs to jump a logarithmic (with respect to the length of the path) number of times, yielding a polynomial runtime.

It is clear that one can create exponentially long paths that Algorithm 1, best and better response dynamics need exponential time to find a Nash equilibrium, but Theorem 4.6 highlights that Algorithm 1 can speed up the convergence to a Nash equilibrium by changing more than one player at a time.

4.3 Finding Nash equilibria with better potential value

Given that our coevolutionary algorithm is able to change more than one player at a time one can conjecture that it can escape from local optima and find Nash equilibria that have a higher potential than the one typically found by best or better response dynamics. Here we prove this is the case for a constructed game (BranchPath): best and better response dynamics fail to find the Nash equilibrium with highest potential value with overwhelming probability and the coevolutionary algorithm finds the Nash equilibrium with highest potential value in expected polynomial time. Although this is a constructed game, we believe similar examples exist, and experiments in Section 5 suggest this behaviour is not uncommon.

The game BranchPath has I players and each player has only two actions (0 or 1). Additionally, all players share the same utility function which will be the potential function.

As potential function, we use a pseudo-Boolean function inspired by a function from [19], considering each bit position in the bitstring as a player. The main idea of the potential function is to create a path with several branches leading towards a local optima. To formally define the function, we use 0^k or 1^k as a substring of k strategies, with only 0 or 1, and $*$ as any strategy.

Definition 4.7. For $I \in \mathbb{N}$, $\sigma = (s_1, s_2, \ldots, s_I)$. Let $s_{\lceil I/2\rceil} := |(s_1, \ldots, s_{\lceil I/2\rceil})|_1$ and $s_{\lfloor I/2\rfloor} := |(s_{I-\lfloor I/2\rfloor}, \ldots, s_I)|_1$ The potential function is defined as:

$$P(\sigma) := \begin{cases} (j+1)I, & \text{if } \sigma = 0^{I-j}\,1^j \\ & \text{with } 0 \leq j \leq \lfloor I/2\rfloor \\ (j+1)I+1, & \text{if } \sigma = 0^{\lceil I/2\rceil}\,1\,0^{\lfloor I/2\rfloor-1-j}\,1^j \\ & \text{with } j \in \{0,2,4,\ldots,2\lfloor(\lfloor I/2\rfloor-1)/2\rfloor\} \\ I - s_{\lceil I/2\rceil}, & \text{if } \sigma = *^{\lceil I/2\rceil}\,0^{\lfloor I/2\rfloor} \\ \lfloor I/2\rfloor - s_{\lfloor I/2\rfloor}, & \text{otherwise.} \end{cases}$$

The "correct" path is the first case of the definition. It starts at the strategy profile 0^I (all 0s) and the next step in the path is found by changing the rightmost strategy to 1, that is, the strategy profile $0^{I-j}1^j$ is the step j in the path. The path ends when the $\lfloor I/2\rfloor$ rightmost strategy are 1 and for all steps in the path the $\lceil I/2\rceil$ leftmost strategies are always 0. Every two steps there is a branch that leads towards a local optima in $0^{I/2}\,1\,0^{I/2-1-j}\,1^j$ for $j \in \{0,2,4,\ldots,\lfloor I/2\rfloor\}$. Outside of the path and the branches, there is a "gradient" towards the start of the path; first to $*^{\lceil I/2\rceil}\,0^{\lfloor I/2\rfloor}$ and then to 0^I. In Figure 1 we show the path and its branches for $I = 10$.

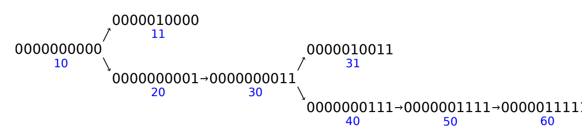


Figure 1: Strategy profiles in the main path and the branches with its potential value in blue.

The highest potential value is at the end of the path ($0^{\lceil I/2\rceil}\,1\,0^{\lfloor I/2\rfloor}$), therefore this is the Nash equilibrium.

We now show that best/better response dynamics find a sub-optimal Nash equilibrium with high probability.

Theorem 4.8. *Best or better response dynamics with random initialisation and Bernoulli random revision sequences does not find the Nash equilibrium σ^* with highest potential value on BranchPath with probability $1-2^{-\Omega(I)}$.*

Similar to the proof of Theorem 4.5 the proof of Theorem 4.8 relies on showing that the algorithms will start the path from the all zero strategy profile with high probability. Then, every two improvement steps in the path there is 1/2 probability that the algorithm gets stuck in a sub-optimal Nash equilibrium, yielding the statement.

Proof. The proof relies on two main ideas. First, with overwhelming probability the algorithm needs to go through the main path from start to finish and second, every two steps of the path there is a 1/2 probability that the algorithm gets stuck in a sub-optimal Nash equilibrium. Now we show these ideas rigorously.

By Chernoff bounds during the initialisation the probability of the initial strategy profile does not have the first $\lceil I/2\rceil$ players use as strategy 0 is $1-2^{-\Omega(I)}$. Therefore, with the same probability the algorithm starts in the fourth case of Definition 4.7. Since the potential value does not increase by changing the strategy of any of the first $\lceil I/2 \rceil$ players and there is at least one player with a strategy 1 there is no player that can unilaterally change its strategy to reach a strategy profile in the first three cases until all the players in the $\lfloor I/2 \rfloor$ positions have a strategy 0. Once a profile strategy with the form $*^{\lceil I/2 \rceil}\, 0^{\lfloor I/2 \rfloor}$ is reached once again there is no player that can unilaterally change its strategy to reach a strategy profile in the first two cases until all the players have the strategy 0. Therefore, the algorithm reaches the main path for the first time with the strategy profile 0^I with probability $1 - 2^{-\Omega(I)}$.

Once this happens for every strategy profile with the form $0^{I-j}\, 1^j$ for $j \in \{0, 2, 4, \dots\}$ there are two players that can unilaterally increase their payoff (and the potential value) one of them leads to a Nash equilibrium and the other continues the path. Since there are at least $\lfloor I/4 \rfloor - 1$ of these steps and each time there is a $1/2$ probability to reach a Nash equilibrium that is not σ^*, then with probability $1/2^{\lfloor I/4 \rfloor - 1}$ the algorithm will find σ^*. □

Now we show that Algorithm 1 finds the Nash equilibrium with the best potential value efficiently.

Theorem 4.9. *Let $I \geq 2$ be the number of players. Algorithm 1 with a mutation probability $0 < \chi \leq 0.3336288$ and a population size $350 \ln\left(\frac{3136(I+\lfloor I/2 \rfloor)}{3\chi^2(1-\chi)}\right) \leq \lambda = O(\ln(I))$ needs $O(I^4)$ payoff evaluations in expectation to find the Nash equilibrium σ^* with highest potential value on* BranchPath.

The proof idea is simple. The longest improvement path has a length of $O(I)$, then we can partition the space in $O(I)$ levels and include the non-optimal Nash equilibria in the levels that have the closest strategy profiles. If the algorithm is not in a Nash equilibrium then it can improve the potential value by changing the strategy of exactly one player, otherwise it can improve the potential value by changing the strategy of exactly two players. Both of these events have a probability $\Omega(1/I^2)$ therefore, we can show all the assumptions to use Theorem 4.2.

Proof. We first define the levels to use Theorem 4.2. The first I levels (A_1 to A_I) are the strategy profiles with potential values from 0 to $I-1$ (third and fourth case), afterwards each level A_{I+j} is the step j in the main path ($0^{I-j}\, 1^j$) and if $j \in \{0, 2, \dots\}$ then the level also includes the local optimum $0^{I/2}\, 1\, 0^{I/2-1-j}\, 1^j$.

Then for the first I levels it is sufficient to change one 1-strategy to 0 and not changing any other strategy. This has a probability at least

$$\left(\frac{\chi}{I}\right)^2 \left(1 - \frac{\chi}{I}\right)^{I-2} \geq \left(\frac{\chi}{I}\right)^2 \left(1 - \frac{\chi}{I}\right)^{I} \geq \left(\frac{\chi}{I}\right)^2 (1-\chi).$$

For the strategy profiles in levels A_{I+j} with $0 \leq j \leq \lfloor I/2 \rfloor$ there is a strategy profile in the next level with only one strategy (in the main path) or two different strategies (in a local Nash equilibrium). As before, for the first case the probability of creating a profile strategy in the next level is at least $\left(\frac{\chi}{I}\right)^2 (1-\chi)$ and in the second case it is sufficient to change two 1-strategy to 0 and not changing any other strategy to create a profile strategy in the next level, therefore this event has a probability of at least

$$\left(\frac{\chi}{I}\right)^2 \left(1 - \frac{\chi}{I}\right)^{I-2} \geq \left(\frac{\chi}{I}\right)^2 \left(1 - \frac{\chi}{I}\right)^{I} \geq \left(\frac{\chi}{I}\right)^2 (1-\chi).$$

Then, for all levels Assumption 1 is met with $(\chi/I)^2(1-\chi)$.

As in the proof of Theorem 4.6 Assumption 2 and Assumption 3 are met given $0 < \chi \leq 0.3336288$ and the population size $\lambda \geq 350 \ln\left(\frac{3136(I+\lfloor I/2 \rfloor)}{3\chi^2(1-\chi)}\right)$. Therefore, by Theorem 4.2 the expected number of evaluations is at most

$$400I \sum_{j=1}^{I+\lfloor I/2 \rfloor} \left(\lambda \ln\left(\frac{58.8I^2\lambda}{196I^2 + 0.8\chi^2(1-\chi)\lambda}\right) + \frac{49I^2}{4\chi^2(1-\chi)}\right) = O(I^4).$$

□

5 Experiments

In this section we analyse whether our results comparing NCP-CoEA and better response (BR) hold for a common game class and whether NCP-CoEA can outperform other common algorithms on a more general and practical application. We first explore settings related to the runtime analysis: we test BR and NCP-CoEA on randomly generated games from the class of Singleton Congestion Games (SCG) with non-monotonic cost functions and show that the NCP-CoEA outperforms BR[3]. Afterwards, we compare our algorithm with other common algorithms on Dynamic Routing Games and show that NCP-CoEA outperforms fictitious play, counterfactual regret minimisation (CFR), and external sampling CFR. For details on the implementation, problem and algorithm settings we refer the reader to Appendix B.

Fig. 2 shows the results for BR and NCP-CoEA on non-monotonic SCG[4]. We observe that BR converges the fastest, likely because NCP-CoEA uses a population that requires more evaluations to update. However, the population in NCP-CoEA helps achieve better Nash equilibria: NCP-CoEA has significantly lower potential values than BR.

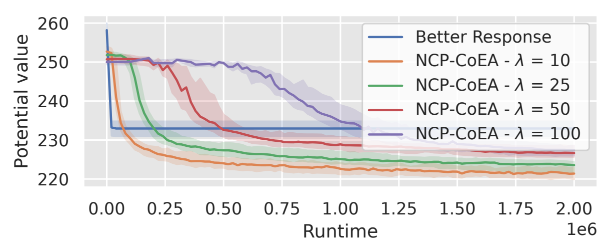
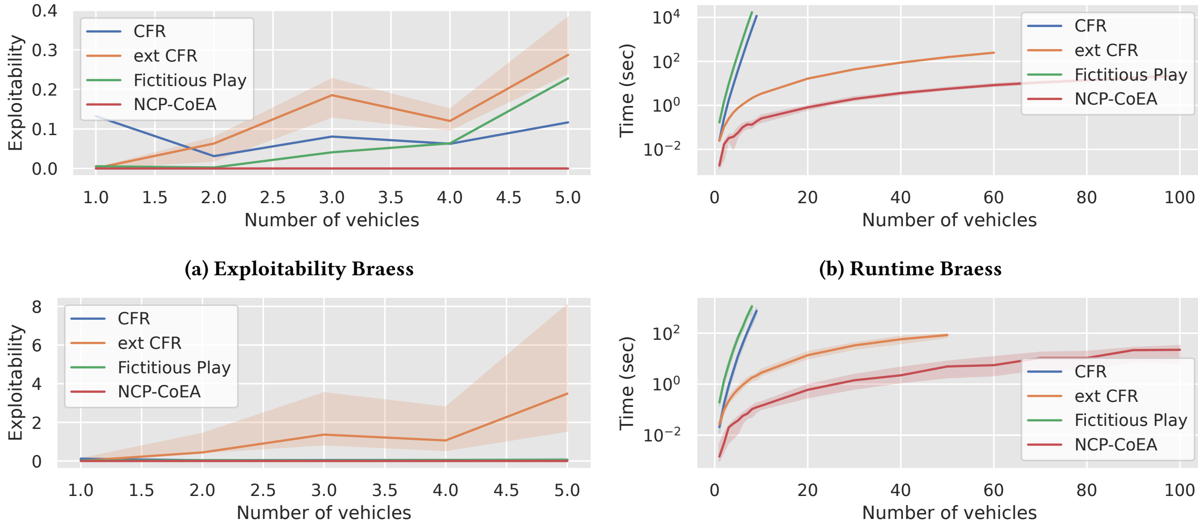


Figure 2: Potential value for random non-monotonic SCG instances with $I = 500, k = 50$. Y-axis shows the potential value. X-axis shows the runtime (oracle calls). Lines show the median best individual with interquartile ranges for the algorithm variants.

Figure 3 shows the results for fictitious play, CFR, external sampling CFR and NCP-CoEA on I-Player dynamic routing games with congestion on the Braess and Pigou networks[5] I-Player dynamic routing games are used because they can be applied to transportation networks [4].

We observe that NCP-CoEA has the lowest exploitability for all networks for all settings, see Figures 3a and 3c. In addition,

[3]Other instances and monotonic SCG are shown in Appendix D.
[4]See Appendix C for details on statistical significance.
[5]See Appendix E for results on laanother network.



(a) Exploitability Braess

(b) Runtime Braess

(c) Exploitability Pigou

(d) Runtime Pigou

Figure 3: Exploitability and runtime for different dynamic routing game networks. The y-axis shows the exploitability or runtime in seconds. The x-axis shows the number of vehicles. The lines show the median best individual with interquartile ranges for the algorithm variants.

Figures 3b and 3d show the results for the runtime. We observe that NCP-CoEA has the shortest runtime, which allows it to solve larger problems.

6 Conclusion

We provided the first theoretical analysis of a coevolutionary algorithm applied to level games, a class of games that includes exact potential games, some weighted and ordinal potential games and certain non-potential games. We established a general technique to obtain upper bounds for the runtime of this algorithm that holds across all games in this important class. Using this technique, we analysed singleton congestion games, giving a polynomial runtime guarantee. Notably, we also showed that there exist games where coevolutionary algorithms can identify Nash equilibria exponentially faster than best response dynamics or better response dynamics. Moreover, the algorithms can find superior Nash equilibria compared to those identified by these approaches. Experimental evaluations further validate that the coevolutionary algorithm exhibits a notable advantage not only with BR on non-monotonic singleton congestion games but also fictitious play, CFR, and external sampling CFR on dynamic routing games consistently finding statistically better Nash equilibria than BR and statistically faster than the other algorithms.

While our results are focused mainly on potential games, we believe that the algorithm presented here can be useful for other non-potential games such as near potential games. For future work we plan to test this hypothesis on more complex games that are not necessarily potential games.

Acknowledgments

Lehre and Hevia Fajardo were 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] Andrea Arcuri and Xin Yao. 2008. A novel co-evolutionary approach to automatic software bug fixing. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. 162–168.

[2] Alistair Benford and Per Kristian Lehre. 2024. Runtime Analysis of Coevolutionary Algorithms on a Class of Symmetric Zero-Sum Games. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Melbourne, Australia) *(GECCO '24)*. Association for Computing Machinery, New York, NY, USA, 9 pages.

[3] Rodica Brânzei, Lina Mallozzi, and Stef Tijs. 2003. Supermodular games and potential games. *Journal of Mathematical Economics* 39, 1 (2003), 39–49.

[4] Theophile Cabannes, Mathieu Laurière, Julien Perolat, Raphael Marinier, Sertan Girgin, Sarah Perrin, Olivier Pietquin, Alexandre M Bayen, Eric Goubault, and Romuald Elie. 2022. Solving N-Player Dynamic Routing Games with Congestion: A Mean-Field Approach. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 1557–1559.

[5] Ozan Candogan, Asuman Ozdaglar, and Pablo A. Parrilo. 2013. Dynamics in near-potential games. *Games and Economic Behavior* 82 (2013), 66–90.

[6] Daizhan Cheng and Zhengping Ji. 2022. Weighted and near weighted potential games with application to game theoretic control. *Automatica* 141 (2022), 110303.

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

[8] Wojciech M. Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. 2020. Real World Games Look Like Spinning Tops. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 17443–17454.

[9] Duc-Cuong Dang, Anton Eremeev, and Per Kristian Lehre. 2021. Escaping Local Optima with Non-Elitist Evolutionary Algorithms. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 14 (May 2021), 12275–12283.

[10] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. 2004. The complexity of pure Nash equilibria. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing* (Chicago, IL, USA) *(STOC '04)*. Association for Computing

Machinery, New York, NY, USA, 604–612.

[11] David González-Sánchez and Onésimo Hernández-Lerma. 2016. A survey of static and dynamic potential games. *Science China Mathematics* 59, 11 (Nov. 2016), 2075–2102.

[12] Qiang He, Cheng Wang, Guangming Cui, Bo Li, Rui Zhou, Qingguo Zhou, Yang Xiang, Hai Jin, and Yun Yang. 2022. A Game-Theoretical Approach for Mitigating Edge DDoS Attack. *IEEE Transactions on Dependable and Secure Computing* 19, 4 (2022), 2333–2348.

[13] T. Heikkinen. 2006. A potential game approach to distributed power control and scheduling. *Computer Networks* 50, 13 (2006), 2295–2311.

[14] Mario Alejandro Hevia Fajardo and Per Kristian Lehre. 2023. How Fitness Aggregation Methods Affect the Performance of Competitive CoEAs on Bilinear Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lisbon, Portugal) *(GECCO '23)*. Association for Computing Machinery, New York, NY, USA, 1593–1601.

[15] Mario Alejandro Hevia Fajardo, Per Kristian Lehre, and Shishen Lin. 2023. Runtime Analysis of a Co-Evolutionary Algorithm: Overcoming Negative Drift in Maximin-Optimisation. In *Proceedings of the ACM/SIGEVO Conference on Foundations of Genetic Algorithms* (Potsdam, Germany) *(FOGA '23)*. Association for Computing Machinery, New York, NY, USA, 73–83.

[16] Yusuke Hino. 2011. An improved algorithm for detecting potential games. *International Journal of Game Theory* 40, 1 (2011), 199–205.

[17] Jeffrey Horn, David E. Goldberg, and Kalyanmoy Deb. 1994. Long path problems. In *Parallel Problem Solving from Nature — PPSN III*, Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 149–158.

[18] Samuel Ieong, Robert McGrew, Eugene Nudelman, Yoav Shoham, and Qixiang Sun. 2005. Fast and Compact: A Simple Class of Congestion Games. In *Proceedings of National Conference on Artificial Intelligence (AAAI)* (proceedings of national conference on artificial intelligence (aaai) ed.). American Association for Artificial Intelligence.

[19] Thomas Jansen, Kenneth A. De Jong, and Ingo Wegener. 2005. On the Choice of the Offspring Population Size in Evolutionary Algorithms. *Evolutionary Computation* 13, 4 (2005), 413–440.

[20] Per Kristian Lehre. 2022. Runtime Analysis of Competitive Co-Evolutionary Algorithms for Maximin Optimisation of a Bilinear Function. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 1408–1416.

[21] Per Kristian Lehre, Mario Hevia Fajardo, Erik Hemberg, Jamal Toutouh, and Una-May O'Reilly. 2023. Analysis of a Pairwise Dominance Coevolutionary Algorithm And DefendIt. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '23)*. Association for Computing Machinery, New York, NY, USA, 9 pages.

[22] Per Kristian Lehre and Shishen Lin. 2024. Concentration Tail-Bound Analysis of Coevolutionary and Bandit Learning Algorithms. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 6940–6948. doi:10.24963/ijcai.2024/767 Main Track.

[23] Per Kristian Lehre and Shishen Lin. 2024. Overcoming Binary Adversarial Optimisation with Competitive Coevolution. In *Parallel Problem Solving from Nature – PPSN XVIII*, Michael Affenzeller, Stephan M. Winkler, Anna V. Kononova, Heike Trautmann, Tea Tušar, Penousal Machado, and Thomas Bäck (Eds.). Springer Nature Switzerland, Cham, 117–132.

[24] Per Kristian Lehre and Phan Trung Hai Nguyen. 2021. Runtime Analyses of the Population-Based Univariate Estimation of Distribution Algorithms on LeadingOnes. *Algorithmica* 83, 10 (Oct. 2021), 3238–3280.

[25] Per Kristian Lehre and Xiaoyu Qin. 2024. More Precise Runtime Analyses of Non-elitist Evolutionary Algorithms in Uncertain Environments. *Algorithmica* 86, 2 (2024), 396–441.

[26] Lina Mallozzi. 2013. An application of optimization theory to the study of equilibria for games: a survey. *Central European Journal of Operations Research* 21, 3 (Sept. 2013), 523–539.

[27] Dov Monderer and Lloyd S. Shapley. 1996. Fictitious Play Property for Games with Identical Interests. *Journal of Economic Theory* 68, 1 (1996), 258–265.

[28] Dov Monderer and Lloyd S. Shapley. 1996. Potential Games. *Games and Economic Behavior* 14, 1 (1996), 124–143. doi:10.1006/game.1996.0044

[29] Una-May O'Reilly, Jamal Toutouh, Marcos Pertierra, Daniel Prado Sanchez, Dennis Garcia, Anthony Erb Luogo, Jonathan Kelly, and Erik Hemberg. 2020. Adversarial genetic programming for cyber security: A rising application domain where GP matters. *Genetic Programming and Evolvable Machines* 21 (2020), 219–250.

[30] Ioannis Panageas, Nikolas Patris, Stratis Skoulakis, and Volkan Cevher. 2023. Exponential Lower Bounds for Fictitious Play in Potential Games. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 32401–32423.

[31] Robert W Rosenthal. 1973. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory* 2 (1973), 65–67.

[32] Günter Rudolph. 1996. How Mutation and Selection Solve Long-Path Problems in Polynomial Expected Time. *Evolutionary Computation* 4, 2 (1996), 195–205.

[33] Karl Sims. 1994. Evolving 3D Morphology and Behavior by Competition. *Artificial Life* 1, 4 (1994), 353–372.

[34] Mark Voorneveld, Peter Borm, Freek Van Megen, Stef Tijs, and Giovanni Facchini. 1999. Congestion Games and Potentials Reconsidered. *International Game Theory Review* 01, 03n04 (1999), 283–299.

[35] Xuehe Wang, Nan Xiao, Tichakorn Wongpiromsarn, Lihua Xie, Emilio Frazzoli, and Daniela Rus. 2013. Distributed consensus in noncooperative congestion games: An application to road pricing. In *2013 10th IEEE International Conference on Control and Automation (ICCA)*. 1668–1673.

[36] Wei Zhong, Youyun Xu, and Huaglory Tianfield. 2011. Game-Theoretic Opportunistic Spectrum Sharing Strategy Selection for Cognitive MIMO Multiple Access Channels. *IEEE Transactions on Signal Processing* 59, 6 (2011), 2745–2759.

[37] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2007. Regret Minimization in Games with Incomplete Information. In *Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis (Eds.), Vol. 20. Curran Associates, Inc.