POLITECNICO DI TORINO
Repository ISTITUZIONALE

Exploiting Evolutionary Modeling to Prevail in Iterated Prisoner's Dilemma Tournaments

Original
Exploiting Evolutionary Modeling to Prevail in Iterated Prisoner's Dilemma Tournaments / Marco, Gaudesi; Piccolo, Elio; Squillero, Giovanni; Alberto, Tonda. - In: IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES. - ISSN 1943-068X. - STAMPA. - 8:3(2016), pp. 288-300. [10.1109/TCIAIG.2015.2439061]

Availability:
This version is available at: 11583/2622370 since: 2016-09-20T11:05:01Z

Publisher:
IEEE

Published
DOI:10.1109/TCIAIG.2015.2439061

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

29 June 2024

Exploiting Evolutionary Modeling to Prevail in Iterated Prisoner's Dilemma Tournaments

Marco Gaudesi, Elio Piccolo, Giovanni Squillero, *Senior Member, IEEE,* Alberto Tonda

***Abstract*—The *iterated prisoner's dilemma* is a famous model of cooperation and conflict in game theory. Its origin can be traced back to the Cold War, and countless strategies for playing it have been proposed so far, either designed by hand or automatically generated by computers. In the 2000s, scholars started focusing on *adaptive players*, that is, able to classify their opponent's behavior and adopt an effective counter-strategy. The player presented in this paper, pushes such idea even further: it builds a model of the current adversary from scratch, without relying on any pre-defined archetypes, and tweaks it as the game develops using an evolutionary algorithm; at the same time, it exploits the model to lead the game into the most favorable continuation. Models are compact non-deterministic finite state machines; they are extremely efficient in predicting opponents' replies, without being completely correct by necessity. Experimental results show that such player is able to win several one-to-one games against strong opponents taken from the literature, and that it consistently prevails in round-robin tournaments of different sizes.**

***Index Terms*—evolutionary algorithms, iterated prisoner's dilemma, non-deterministic finite state machine, opponent modeling.**

I. Introduction

THE prisoner's dilemma is a renowned model of cooperation and conflict in game theory. In its simplest form, it describes an interaction between two players whose final outcome is determined by the choices of both. There are only two possible moves: C and D. If both players choose C, they earn a high reward; if both play D, they get only a low reward. If one chooses C and the other D, the latter obtains a very high reward, while the former endures a very low reward or no reward at all [**?**]. It is a *nonzero-sum game*, because whatever benefit accrues to one player does not necessarily imply the same penalty imposed on the other player. The model was originally devised in 1950 by the mathematicians Merrill Flood and Melvin Dresher for simulating world leaders' struggle between peaceful coexistence and nuclear aggression. The study was part of the RAND[1] Corporation's research into game theory and its applications to global nuclear strategy [**?**].

To make the idea more accessible to psychologists, a few years later Albert Tucker labeled the problem "Prisoner's Dilemma" and described it with the following metaphor: two criminals are arrested for an offense and placed in separate isolation cells; each of two prisoners is offered freedom in exchange for implicating the other. If neither does, they will receive the usual sentence; however, if both prisoners accuse each other, they are granted moderately harsh sentences.

In such metaphor, negating the involvement is "cooperating", corresponds to move C, and represents the selfless and peaceful behavior; betraying the ex-partner is "defecting", corresponds to D, and symbolizes the selfish and aggressive behavior. That is, the most favorable situation is to confess while the partner is negating any involvement, and the least favorable one is to be betrayed while resisting the police's pressure.

Let $\mathbf{R}$ (*reward*) be the payoff for a mutual cooperation, and $\mathbf{P}$ (*punishment*) the payoff if both players defect; when only one player acts selfishly, its payoff is $\mathbf{T}$ (*temptation*), and the payoff of its opponent is $\mathbf{S}$ (*sucker*). The Prisoner's Dilemma model requires that[2]:

$$\mathbf{T} > \mathbf{R} > \mathbf{P} > \mathbf{S} \tag{1}$$

$$\mathbf{R} > \frac{\mathbf{S} + \mathbf{T}}{2} \tag{2}$$

and the most common payoff matrix found in literature sets $\mathbf{T} = 5$, $\mathbf{R} = 3$, $\mathbf{P} = 1$, and $\mathbf{S} = 0$.

According to equation (1), in a single turn of the prisoner's dilemma the selfish move always dominates the selfless one: if one of the players cooperates, the most profitable decision for the other is to exploit his good faith and defect; on the other hand, if a player defects, the best counteraction for the opponent is, again, to be selfish and minimize the damage. Therefore, the situation has no strategic interest. Similarly, using the technique of *backward induction*, it may be inferred that the optimal strategy in a game of known length is to always defect [**?**].

Quite differently, in the *iterated prisoner's dilemma* (IPD) two players face the previously described situation for an unknown, potentially infinite number of turns with memory of the previous exchanges. Here the model changes significantly: repeatedly defecting is definitely not the optimal strategy, and equation (2) discourages from alternating between cooperation and defection. As a matter of fact, the single-shot game is so uninteresting, that frequently in literature the term "prisoner's dilemma" refers to the iterated version.

Authors are listed in alphabetical order.

Marco Gaudesi, Elio Piccolo, and Giovanni Squillero are with Politecnico di Torino, Torino, Italy.

Alberto Tonda is with UMR 782 GMPA, INRA, Thiverval-Grignon, France.

[1]*Research ANd Development*, a nonprofit global policy think tank formed to offer research and analysis to the United States armed forces.

[2]In game theory, *payoffs* are positive and the objective is to maximize the reward. However, since in Tucker's formulation the final goal was minimizing a prison sentence, equations (1) and (2) can be found reversed in some textbooks.

Despite its simplicity, or perhaps because of its simplicity, the iterated prisoner's dilemma proved to be an excellent model of many real-life situations, from animal behaviors to economic strategies. Scientific contributions on the topic appear regularly in various fields, from theoretical biology to mathematics, game theory, and computer sciences. Moreover, many variants of the basic model appeared and have been studied over the years. For instance: in the *spatial prisoner's dilemma*, individuals could only mate and play with their neighbors in a lattice or other structure [**?**]; in the *evolutionary IPD*, a population of strategies plays IPD with one another and each strategy has the chance to produce offspring proportional to its payoff [**?**]; moreover the idea of a topological structure may be used in the evolutionary analysis [**?**]. In the *N-person prisoner's dilemma*, the game is extended to multiple players [**?**]; while in the *shopkeeper* model, a player interacts with a series of opponents without resetting its internal state, simulating a shopkeeper that interacts with a series of customers [**?**].

Robert Axelrod, a political scientist at the University of Michigan, was the first to attempt searching for efficient strategies using competitions. In 1979 he organized an IPD tournament soliciting strategies from game theorists who already published in the field, and, soon after, ran a second round of the same tournament with a slightly modified setup [**?**]. Since then, new tournaments have been held regularly. Interestingly, while original works aimed at studying the emerging cooperation, after three decades scholars are also focusing on the *emerging exploitation*, that is, strategies that succeed at the expense of their opponents [**?**].

Defining a fixed strategy able to overcome all possible opponents in such tournaments has been demonstrated impossible. Some researchers tried to find strategies averagely effective against a varied set of adversaries; others developed strategies able to recognize and team-up with member of the same kin. Scholars also proposed adaptive strategies able to shape their behavior according to the opponent's actions. Some adaptive players record a statistic of their opponent's moves to find the most profitable action: a first-order distribution probability is often used to determine the expected payoff for defection and cooperation, possibly smoothed with heuristics. Other adaptive players try to categorize the opponent using rule-based mechanisms to play the best counter strategy [**?**].

In a recent line of research, the idea of categorizing the opponent was pushed further: instead of choosing from a set of predetermined archetypes, the player builds a model of the opponent from scratch using evolutionary computation. The heuristic optimizer is used not to generate a strategy before the game, but to tweak the strategy during the game. This idea was tested with two players: *Laran*[3] in 2011 [**?**]; and *Turan*[3] in 2014 [**?**].

Building on these experiences, this paper presents *Tages*[3], an evolutionary player that tries to predict the opponent's future moves in order to select the most profitable line of play. Tages models the opponent's strategy as a non-deterministic finite automaton; considering the record of past moves and simple heuristics, it is able to build effective representations. Tages requires no training nor preparation to face new opponents: the internal model is built ab initio and tweaked as the game develops.

Experimental results show that the proposed methodology is able to win standard round-robin tournaments, outperforming both deterministic and non-deterministic strategies. Moreover, the models are highly functional as their predictions are fulfilled more than 90% of the times against most opponents.

The rest of the paper is organized as follows: section II contains a brief summary of relevant strategies developed for past tournaments; section III gives an overview of the proposed approach, while sections IV and V detail the modeling and the playing units; section VI shows the experimental evaluaton; finally, section VII concludes the paper.

II. IPD Strategies

Back in 1979, during the first tournament organized by Robert Axelrod, each player was matched against all others in round-robin over a sequence of 200 turns, and the total payoff collected during all games was considered as the final score. Fourteen strategies were submitted, and Axelrod himself added a fifteenth strategy that cooperated and defected with equal probability. The competition was won by *Tit for Tat*, a deceptively simple strategy created by the Russian-born American mathematical psychologist Anatol Rapoport. *Tit for Tat* cooperates on the first turn and then mimics the opponent's choice in the previous turn.

Soon after, Axelrod ran a second round of the tournament. He adopted the same round-robin scheme, but, to prevent the use of backward induction, the length of games was not disclosed in advance. On the contrary, he set a probability $p = 0.00346$ for each turn to conclude the game. Sixty-two entries were submitted, and *Tit for Tat* claimed the first place once again [**?**].

Tit for Tat was the simplest strategy submitted to the tournament, and it won both rounds. Even though all the participants perfectly knew who had won the first round, no one was able to design an entry that did any better in the second. Under a closer examination, this is not surprising: *Tit for Tat* never defects first, it is never taken advantage of more than once without retribution, and it is swift in forgiving. Rapoport's strategy soon became an important point of reference in IPD studies, both to evaluate and to classify new strategies. Being able to outperform *Tit for Tat* in a tournament is still a challenge, as a player specifically tweaked to overcome it is likely to lose many points against the other participants.

Over the years, new players for the IPD have been both carefully crafted by hand, and automatically generated by machines. Their descriptions are scattered through scientific literature and dedicated Internet web sites[4]. Among the many different approaches to automatically devise effective strategies, *genetic algorithms* have been exploited since the very beginning [**?**]. Scientific literature also reports the use of

[3]All names are taken from the Etruscan mythology: Laran is the god of war, Turan his consort, and Tages a deity gifted with prescience.

[4]See for example http://www.prisoners-dilemma.com/strategies.html

several nature-inspired heuristics, such as *evolutionary programming* [**?**], *particle swarm optimization* [**?**] or *artificial neural networks* [**?**]. An insightful survey on evolutionary-based approaches can be found in [**?**], but the field is very active and new contributions are appearing regularly. More conventional techniques, such as *Q-Learning* [**?**], have also been applied to the task.

Finite state machines (FSMs) have been commonly employed to represent strategies for the IPD. They are a compromise between simplicity and expressiveness, as they can represent very complex Markov models [**?**]. Most of the works use *Moore machines*, i.e., FSMs whose output value is determined by the current state, only, while a few utilized *Mealy machines*, where the output is a function of the current state and input[5]. In modern strategies, however, the output is not always a deterministic function of the input because players make use of random number generators. Moreover, whenever strategies exploit lookup tables, counters, or floating-point variables, FSMs become, if not inadequate, at least unpractical, because the number of states would be enormous.

While facing the problem of creating effective strategies, scholars also tackled the problem of classifying strategy's behavior. In 2000s, *fingerprinting*[6] has emerged as an effective analysis tool independent from the representation [**?**] [**?**]. A more comprehensive survey on the topic may be found in Jeffrey Tsang's 2014 Ph.D. dissertation[7]. Common strategies are here classified informally into four, intuitive broad groups. For the detailed description of each one see Appendix A.

1) Dummy Strategies: This group contains trivial players that are commonly considered as a testbed to assess the efficacy of complex strategies. For instance, strategies that blindly repeat a fixed sequence of moves, or strategies that base their choice on pure randomness. Remarkably, while these strategies are trivial to describe, adding them to a tournament may bias the results, and finding an optimal strategy against all of them is not always straightforward.

2) Tit for Tat and Derivative Strategies: This group contains the original *Tit for Tat* and other strategies that have been directly derived from it. It also includes *Reverse Tit for Tat*, a player that retaliates by cooperating if the opponent defected in the previous turn.

3) Group Strategies: Group strategies assume the presence of other players of the same kin in the tournament. They aim at recognizing each other and increase their payoff [**?**]. If the opponent is not identified as a member of the team, sometimes they play as *Tit for Tat*, or sometimes as *Always Defect*.

One famous example is the strategy devised by researchers at the Southampton School of Electronics and Computer Science: they submitted to the same tournament over 60 players able to recognize each other with an initial handshake and then collude, raising the score of selected members of their team, while lowering the score of other players with endless defections. As a result, members of the squad gained the first three positions in the tournament[8].

4) Adaptive Strategies: While most of the players do take into account the opponent's behavior, players in this group are able to drastically change their strategy as a result of an explicit evaluation of the opponent. Several strategies in this group also stem from *Tit for Tat*, like *Omega Tit for Tat* [**?**]. A truly remarkable example is Jiawei Li's *Adaptive Pavlov* [**?**], that uses a rule-based mechanism to switch between three possible behaviors, namely *Tit for Tat*, *Soft Tit for Two Tats*, and *Always Defect*.

An idea common to many adaptive strategies is to *explore* and *exploit*. In the former phase the opponent is tested, while in the second the best counter strategy is adopted. Usually the first phase lasts a predetermined number of turns, and may be repeated cyclically or upon a drop in performance. Since the 2000s, such adaptive strategies consistently ranked very high in IPD tournaments [**?**], [**?**].

5) Zero-determinant Strategies: Zero-determinant (ZD) strategies entered the scene of the IPD in 2010s when William Press and Freeman Dyson showed the unexpected existence of *ultimatum strategies* [**?**]. Such strategies impose an ultimatum to the opponent, forcing it to concede points with the threat of defection. They impose a certain relationship between their score and the opponent's, first choosing their target objective, and consequently deriving their probabilities of cooperating or defecting.

ZD strategies were proved to be evolutionary unstable by Christoph Adami and Arend Hintze [**?**]; few months later, however, Alexander Stewart and Joshua Plotkin demonstrated that the subset labeled "generous ZD strategies" is stable in large populations [**?**].

III. Proposed Approach

This paper presents Tages, an adaptive IPD player based on opponent modeling. Its goal is to take the most of its adversary by leading the game into a profitable line of exchanges. To this end, it builds a model of the strategy it is facing, although the primary goal for such a model is to be functional rather to be exact: Tages does not need to model an opponent exactly in order to deal with it optimally.

Tages is conceptually composed of two distinct blocks: a *modeling unit*, based on an evolutionary optimizer; and a *playing unit*, based on a brute-force algorithm (Fig. 1). Each turn, the modeling unit updates a coherent, competitive and compact model of the opponent. The process is not based on any assumptions about the adversary, such as a list of established strategies to choose from, but creates the model from scratch. Then, the playing unit retrieves the model, replays the current game, simulates all possible 2^M continuations of M moves, and selects the one that provides the highest payoff. The playing unit eventually selects the first move in this sequence. In rare occasions, however, it rather exploits simple heuristics instead of the brute force analysis (see Section V for details).

[5]Moore and Mealy machines are equivalent formalism: it is trivial to translate a FSM from one representation to the other by adding or removing states.

[6]A treatise of fingerprinting can be found in Eun-Youn Kim's Ph.D. dissertation: E. Y. Kim. *Fingerprinting: Automatic Analysis of Evolved Game Playing Agents*. Ph.D. Thesis, Iowa State University, 2005

[7]J. Tsang. *The Structure of the Space of All Game-Playing Strategies*. Ph.D. Thesis, University of Guelph, 2014

[8]*University of Southampton team wins Prisoner's Dilemma competition* http://www.southampton.ac.uk/mediacentre/news/2004/oct/04_151.shtml

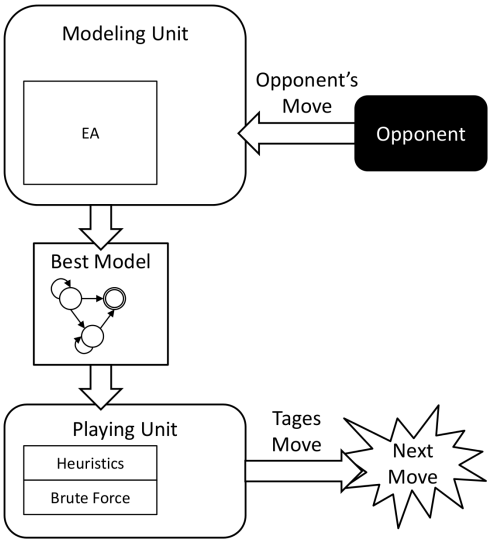

Fig. 1. The general architecture of Tages.

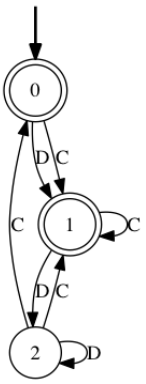

Fig. 2. An example of ND-FSM. States are numbered starting from 0. The output is "cooperation" (C) in states drawn with a double border, and "defection" (D) otherwise. The letter on each transition indicates the triggering opponent's move. The bold arrow with no label and no antecedent points to the initial state. This ND-FSM is a model of *Generous Tit-for-Tat* built by Tages during a game: nodes 1 and 2 encode the standard *Tit-for-Tat*, while node 0 enables it not to retaliate at once. Evidently, during the game it forgave Tages' first deception but not the second.

As said before, it may not be practical to describe a complex strategy with a simple FSM. Tages, therefore, models its opponent as a *non-deterministic finite state machine* (ND-FSM), sometimes called *non-deterministic finite automata*[9]. In a model, transitions are always triggered by Tages' move, but the same input symbol may cause a transition to two or more states. Such non-determinism is used to express uncertainty in a compact form: the opponent sometimes behaves in a certain way, sometimes behaves differently. Additionally, the initial state might not be unique, as different behaviors can be recorded from the very first turns[10]. See Fig. 2 for an example, and the explanation of the keys used in the rest of the paper.

In the text, the following terminology is adopted: the sequence of moves played by Tages defines the *path*, and the moves performed by opponents are the *footprints* left while walking on the path, each turn representing a single *step*. Considering the game record, the footprints are the sequence of moves played by the actual opponent. While modeling an opponent, differently, the footprints left by a candidate model are all the possible sequences of moves that it could have played given Tages' fixed ones. Thus, models are likely to leave footprints composed of more and more sequences as their non-determinism increases.

IV. Modeling Unit

Tages exploits an evolutionary algorithm (EA) to build and optimize the model of the opponent's behavior. The EA handles a population of μ individuals, each one a candidate model encoded as a ND-FSM. Each generation, the EA goes through the standard phases: *reproduction*, the creation of offspring; *evaluation*, the evaluation of new individuals; *removal*, the removal from the population of less fit individuals. The optimization process is stopped after $\mathcal{G}$ generations, or after $\frac{1}{2}\mathcal{G}$ generations without improvement.

A. Initialization

Before the first generation, 5μ individuals are created from scratch: they are made of exactly ι states, their outputs and the transitions between them are randomly chosen. Then, with a small probability, additional transitions are added, possibly generating a non-deterministic strategy. Eventually, the most competitive and compact μ are kept as the initial population (see IV-E for details on how individuals are evaluated).

B. Reproduction

In the reproduction phase, the size of the population is doubled creating μ new individuals. With probability $p = \frac{1}{4}$ Tages generates the offspring through recombination of two parents; with probability $p = \frac{3}{4}$ through the mutation of a single parent (for details on the selection mechanism, see IV-F).

There is only one recombination operator that simply juxtaposes the ND-FSMs of the two parents, maintaining the initial states of both. As a result, the offspring may behave either as one parent or as the other. There are nine mutation operators, and Tages selects which one to apply with a probability biased by their performance, that is, the number of individuals in the current population that have been created by that operator. Individuals with no parents are considered created by all operators, so, in the first generation, all selection probabilities are the same.

In more details, the nine mutation operators may: change one of the initial states of the ND-FSM; add a non-initial state to the set of initial states – thus increasing the overall non-determinism; remove a state from the set of initial states –

[9]A ND-FSM is a FSM that (i) does not necessarily require input symbols for state transitions and/or (ii) is capable of transitioning to two or more states for a given start state and input symbol. ND-FSM and FSM are equivalent formalisms [**?**] [**?**].

[10]A ND-FSM with multiple initial states can be easily translated to a ND-FSM with single initial state.

thus decreasing the overall non-determinism; add a state with a random output and two random transitions, then connect the state to the ND-FSM adding a new non-deterministic transition from a random preexisting state – thus increasing the overall non-determinism; remove a state along with all its ingoing/outgoing transitions; change the destination state of a transition; switch the output of a state from "cooperate" to "defect" or vice-versa; add an outgoing transition to a state – thus increasing the overall non-determinism; remove an outgoing transition from a state which has two or more transitions triggered by the same event – thus decreasing the overall non-determinism.

There are several operators that increase the size in the offspring, as there are several different ways to do it, but bigger models are likely to receive lower fitness and self-adapting activation probabilities reduces the bloating phenomenon. Moreover, some mutations may fail, i.e., not be able to produce an offspring. For instance, *RemoveInitialState* is bound to fail if there is only one initial state in the ND-FSM. In such cases, no individuals are added to the population.

C. Extinction

Premature convergence is a taxing problem in evolutionary computation. Moreover, in Tages the fitness is dynamic, and all individuals in the population may suddenly become almost useless because no one is able to model the last move of the opponent. To escape such a situation, the modeling unit makes use of *Extinction*, a well-known technique where a large part of the population is suddenly killed and replaced with new solutions [**?**].

Tages resorts to extinction if it is not able to enhance the best solution for $\frac{1}{3}\mathcal{G}$ generations. When the condition is met, instead of the normal reproduction phase, $\frac{4}{5}$ of the individuals are removed from the current population, and new 5μ individuals are added. Such individuals are created from scratch as before the first generation (see IV-A), consequently, the result is also to reduce the gaps between the different operator activation probabilities.

Finally, all individuals in such an abnormally large population are evaluated, and the best μ preserved for the next generation.

D. Evaluation and Removal

In these phases, Tages first evaluates all new individuals, then removes the less fit from the population until its size reaches back μ.

E. Fitness Function

Candidate models are ranked according to their *coherence* with the game record, general *competitiveness*, and *compactness*. That is, to be functional a model is requested to adhere with the game so far, to describe a strong opponent, and to be general and tractable.

The first requirement is essential for the model to be sensibly used to predict the future continuation of the game. The second requirement was introduced in *Laran* [**?**]: it sounds reasonable to assume that the opponent is strong when devising a strategy against it. The third pushes Tages to create expressive models at the expense of exactness.

1) Coherence: The coherence could be defined as the amount of overlapping between the footprints of the model and the moves recorded during the game. It should measure whether a model can be used to predict the future development of the game, or how far it is from being able to do it. While fingerprinting and multilateration could be theoretically used for the second task [**?**], it would not answer the first question. Moreover an explicit evaluation is not practical as the number of sequences composing the footprints of a ND-FSM increase exponentially over the length of the path. Tages resorts to two measures to approximate an exact coherence called respectively *plausibility* and *likelihood*.

The plausibility ($\mathcal{P}$) measures the tail of the game sequence that can be found in the strategy's footprints. It is a real number, $\mathcal{P} \in [0, 1]$. If the entire sequence is found in the footprints, then $\mathcal{P} = 1$. Otherwise the ND-FSM representing the strategy is simulated starting from the last turn; transitions are played backwards, dropping states with mismatching outputs until no more state is active; and the final value of $\mathcal{P}$ is proportional to the number of steps.

The plausibility does not measure how many sequences in the footprints correspond to the sequence of moves actually played in the game, nor it takes into consideration the total size of the footprints. However, with complexity $\mathbf{O}(s)$ for a game of length s, it assesses whether the considered model could have produced the correct sequence; and if not, it gives a rough approximation of how *close* it is to producing that sequence.

The likelihood ($\mathcal{L}$) of a given model $\boldsymbol{\theta}$ is the opposite of the logarithm of the zero-order probability that in turn i the model plays the same move recorded in the game G_i, summed up over all turns in the game[11]. The likelihood is a positive number, $\mathcal{L} \in [0, +\infty]$, the smaller the better. Steps are weighted to give slightly more relevance to recent moves over the ones played in the beginning of the game ($\xi > 1$):

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_i \xi^i \cdot \log(P(\theta_i = G_i))$$

Where the probability $P(\theta_i = G_i)$ for the model $\boldsymbol{\theta}$ to match the action G_i after the previous $G_1 ... G_{i-1}$ moves is:

$$P(\theta_i = G_i) = \frac{n_{action}}{n_{total}}$$

where n_{action} is the number of current states returning action G_i, while n_{total} is the total number of states the ND-FSM is in at the moment.

The likelihood can be calculated with complexity $\mathbf{O}(s)$ for a game of length s, but it is an imprecise measure since it does not take into consideration sequences. For instance, the footprints of a non-deterministic player that could behave either as an *Always Defect* or as an *Always Cooperate* are composed of two sequences: a sequence of uninterrupted cooperations, and a sequence of uninterrupted defections. However, such

[11] More precisely, this value should be called *log* likelihood [**?**].

a strategy would get exactly the same likelihood value on any path, as the zero-order probabilities for cooperation and defection are always the same ($\forall i : P(\theta_i = G_i) = \frac{1}{2}$).

2) Competitiveness: The competitiveness ($\mathcal{C}$) measures the performances of the strategy against some sparring partners. It is a positive integer number, $\mathcal{C} \in [0, +\infty]$, the larger the better. Tages employs only two adversaries: a player that cooperates with a probability equal to the frequency of cooperations in its opponent record; and a player that behaves like *Tit for Tat* nine times out of ten, and otherwise choses the opposite move.

The ND-FSM describing the model is matched for nine games of length of 100, 200, ..., 900 against each opponent, and the competitiveness is the total payoff earned. Whenever the strategy prescribes more than one transition for the same input symbol, one is chosen randomly with flat probability. As a consequence, games may be non reproducible, and the competitiveness should be considered as a noisy fitness function.

3) Compactness: The compactness ($\mathcal{X}$) takes into account both the size and the amount of non determinism of the ND-FSM. Is is a positive real value, $\mathcal{X} \in [0, +\infty]$, the smaller the better. The individual is simulated using the path as input. For each node that has been visited at least once in the last 20% of the steps, a low penalty ($e = 1$) is considered; for each other node that has been visited at least once in the last 50% of the steps, a medium penalty is considered ($e = 5$); nodes that have been visited only in the first 50% of the path contribute with a high penalty ($e = 10$). A maximum penalty ($e = 20$) is associated to isolated nodes, i.e., nodes that cannot be reached regardless of the path. All penalties factors are summed up and multiplied by the total number sequences in the footprints.

Whether a model has a number of states higher than a threshold ζ, it is considered not usable, regardless its plausibility. After the initial turns, the value of ζ is equal to the percentage of correct predictions achieved by the model: that is, a model that is able to correctly foresee 75% of the opponent's responses is allowed to contain up to 75 states. The rationale is quite simple: if the model is precise, it may be also complex; but if the model is not functional it is better to increase its generality than its size.

F. Selection

In order to be usable, a model must be fully plausible ($\mathcal{P} = 1$) and the number of states in it must be less than ζ. Individuals are chosen for reproduction through tournaments of size τ. First of all, if only one candidate is usable it is preferred. If both or neither are usable, the two candidates are compared on all aspects of the fitness: $\mathcal{P}$, $\mathcal{L}$, $\mathcal{C}$, and $\mathcal{X}$.

Comparison is performed using the *chromatic operator* [**?**]: with a random wheel selection weighted on the relative difference between corresponding fitness values; in other words, fitness components for which the two individual differ the most have a higher probability of being selected for comparison.

V. Playing Unit

The playing unit exploits the best model devised by the modeling unit, and select the next move to be played by Tages. Finally, if the model is reliable, the most probable opponent's reply is stored as the "prediction", and used to calculate statistics (see sections IV-E3 and VI).

A. Brute-force Analysis

First, the current game is replayed up to the current turn to detect the set of possible current states of the ND-FSM. To perform predictions, a model must be fully plausible ($\mathcal{P} = 1$), that is, the set of possible current states cannot be empty.

Then, using a brute force approach, all 2^M possible continuations of M moves are simulated. The simulation takes into account non-determinism: whenever the model may behave differently, the average payoff for the step is used. Then, the sequence yielding the highest payoff is taken, and the first move is selected.

B. Exploration

With a very small probability $p = \theta$, the unit does not play the move selected by the brute-force analysis, but its opposite, to explore the opponent's behavior.

C. Random Move

If the model is not able to replay the current game, that is, $\mathcal{P} < 1$, the unit selects a random move. The probability of choosing "cooperation" is equal to the fraction of the total reward received so far when cooperating.

D. Reconciliation

To break long sequences of mutual defections, there is a probability to attempt a *reconciliation*, that is, to play a sequence of cooperations ignoring opponent's replies. Such probability $p = n \cdot \rho$ is proportional to the length n of the sequence, and a coefficient ρ that is scaled down by ten after each reconciliation attempt. The length of the sequence is selected with uniform probability between 4 and 10.

VI. Experimental Evaluation

Tages was implemented in the *Go* programming language, and consists of about 5,000 lines of code. The source code is publicly available from Bitbucket[12]. The parameters used during the evaluations are shown in Table I.

The effectiveness of Tages was first assessed with testbeds, then its performance in typical tournaments was evaluated. The standard payoff matrix was adopted: $\mathbf{T} = 5$, $\mathbf{R} = 3$, $\mathbf{P} = 1$, and $\mathbf{S} = 0$. In the first sets of experiments, all matches consisted of one game, that lasts exactly 200 turns. When simulating tournaments, each strategy was confronted with all the others in a round-robin fashion. Matches were composed of 5 games of exactly 168, 359, 306, 622, and 319 turns. The lengths, precomputed using Axelrod's rules, are also the default in the well-known software *Oyun* [**?**]. However, no strategy took advantage of this knowledge. Each evaluation has been repeated 10 times with different random seeds.

[12]https://bitbucket.org/squillero/tages

The experiments consider the efficacy of the models, and either the total payoff or the average payoff per turn. For all, both the average over ten runs and the standard deviation is reported. The "efficacy" is measured as the fraction of turns where the prediction is correct over the total. That is, the ratio of turns where the modeling unit yields a model fully coherent with the game played so far, and the very same model is correct in anticipating the next opponent's move (see section V). Thus, values can be as low as zero.

TABLE I
PARAMETERS USED IN TAGES FOR RUNNING ALL THE EXPERIMENTS.

Param	Value	Description
μ	100	Population/offspring size
$\mathcal{G}$	300	Maximum number of generations
θ	0.01	Probability to explore the model
ρ	0.1	Initial reconciliation factor
ι	5	Size of the initial random models

A. Preliminary Assessment

The first set of experiments confronts Tages with players of the *Dummy* group, that is, players that act blindly without considering opponents' moves. Table II shows five players that repeat a predefined sequence, and eleven random players $\mathcal{R}_\delta$ that cooperate with probability $p = \delta$. $\mathcal{R}_{1.00}$ and $\mathcal{R}_{0.00}$ also belong to the first group, and correspond to *Always Cooperate* and *Always Defect*. The player with $\delta = \frac{1}{2}$ defects and cooperates with equal probability, and it is sometimes called *Random*.

For each opponent, the table reports the average payoff earned in each turn and the efficacy of the model over ten runs. For each parameter, both the average value ($\overline{P}$) and the standard deviation (P_σ) are reported. With such simple opponents, it is straightforward to calculate the optimal average payoff per turn (P_{opt}), and the information is shown last in the column block. The third column block reports the efficacy of the model: average ($\overline{E}$) and standard deviation (E_σ).

TABLE II
PERFORMANCE AGAINST "DUMMY" STRATEGIES: AVERAGE PAYOFF PER TURN AND MODEL EFFICACY (AVERAGE AND STANDARD DEVIATION OVER 10 GAMES). OPPONENTS NAMED $\mathcal{R}_p$ PLAY RANDOMLY, COOPERATING WITH PROBABILITY p.

Opponent name	Payoff			Model	
	$\overline{P}$	P_σ	P_{opt}	$\overline{E}$	E_σ
CD	2.91	0.04	3.00	0.99	0.00
CCD	3.57	0.02	3.67	0.97	0.01
DDC	2.19	0.02	2.33	0.97	0.01
$\mathcal{R}_{1.0}$ (AC)	4.98	0.01	5.00	1.00	0.00
$\mathcal{R}_{0.9}$	4.54	0.18	4.60	0.46	0.22
$\mathcal{R}_{0.8}$	3.97	0.26	4.20	0.33	0.18
$\mathcal{R}_{0.7}$	3.43	0.44	3.80	0.12	0.06
$\mathcal{R}_{0.6}$	2.92	0.14	3.40	0.14	0.08
$\mathcal{R}_{0.5}$ (RND)	2.68	0.24	3.00	0.11	0.10
$\mathcal{R}_{0.4}$	2.14	0.17	2.60	0.16	0.04
$\mathcal{R}_{0.3}$	1.83	0.12	2.20	0.14	0.08
$\mathcal{R}_{0.2}$	1.51	0.11	1.80	0.25	0.06
$\mathcal{R}_{0.1}$	1.01	0.15	1.40	0.56	0.05
$\mathcal{R}_{0.0}$ (AD)	0.86	0.03	1.00	1.00	0.00

Since the behavior of such opponents cannot be influenced, the optimal line of action is to always defect. As it could be expected, the accuracy of the predictions is very high when the model is fully deterministic and decreases sharply as randomness increases.

As a matter of fact the performance of Tages are within 10% of the optimum for opponents with $\delta \in]0, 0.5[$, and fall down to -23% for opponents with $\delta \in [0.5, 1[$. The program is far more effective with the first half of players because such players are more likely to get a good score in a generic tournament, and Tages assumes that its opponents are strong.

In the second experiment Tages is confronted with *unreliable Tit-for-Tat* opponents. Such strategies act as the original Rapoport's with probability $p = \alpha$, while they chose the opposite move with probability $p = 1 - \alpha$. As in the previous test bench, the first and last are deterministic, and correspond respectively to *Tit for Tat* and *Reverse Tit for Tat*. Table III reports, for each opponent, the average payoffs earned in each turn, and the efficacy of the model.

TABLE III
PERFORMANCE AGAINST RANDOMIZED TIT-FOR-TAT STRATEGIES: AVERAGE PAYOFFS PER TURN AND MODEL EFFICACY (AVERAGE AND STANDARD DEVIATION OVER 10 GAMES). STRATEGIES PLAY "TIT FOR TAT" WITH PROBABILITY α, AND "REVERSE TIT FOR TAT" OTHERWISE.

α	Tages		Opponent		Model	
	$\overline{P}$	P_σ	$\overline{P}$	P_σ	$\overline{E}$	E_σ
1.0 (TFT)	2.96	0.01	2.96	0.01	0.98	0.00
0.9	2.46	0.14	2.62	0.25	0.65	0.21
0.8	2.21	0.05	1.96	0.17	0.25	0.13
0.7	2.30	0.12	1.85	0.42	0.23	0.13
0.6	2.46	0.16	1.54	0.43	0.11	0.07
0.5	2.63	0.28	1.51	0.54	0.10	0.04
0.4	2.84	0.22	1.30	0.34	0.09	0.04
0.3	3.40	0.21	0.79	0.17	0.08	0.04
0.2	3.71	0.26	0.69	0.26	0.12	0.07
0.1	4.16	0.49	0.46	0.37	0.47	0.28
0.0 (RTFT)	4.92	0.04	0.04	0.03	0.98	0.00

As seen in the previous experiment, Tages is able to build highly efficient models against deterministic players, and it is understandably unable to predict the actions of random ones.

Solving the relevant Bellman equations, it may be proven that the best strategy is *AC* when $\alpha > 0.8$; *CD* when $\alpha \in [\frac{2}{3}, 0.8]$; and *AD* when $\alpha < \frac{2}{3}$ (see Jeffrey Tsang's 2014 Ph.D. dissertation for details[7]). Tages' Performances are nearly optimal against deterministic players, and are generally better when playing against adversaries more similar to *Tit for Tat* than *Reverse Tit for Tat*.

B. Tit for Tat and Derivatives

Table IV reports the results against common players derived from Rapoport's *Tit for Tat* (for details see Appendix A). Differently from the first set, adversaries listed here do have their own "strategy", and the modeling unit can therefore be effectively exploited.

The accuracy of the prediction is linked to the amount of intrinsic randomness of the strategies. Some of these players can be fooled by an alternated sequence of cooperation and defection, such as *TF2T* and *HTF2T*, and Tages exploits this

TABLE IV
PERFORMANCE AGAINST "TIT-FOR-TAT" STRATEGIES: AVERAGE PAYOFFS PER TURN AND MODEL EFFICACY (AVERAGE AND STANDARD DEVIATION OVER 10 GAMES).

Opponent name	Tages		Opponent		Model	
	$\overline{P}$	P_σ	$\overline{P}$	P_σ	$\overline{E}$	E_σ
2TFT	2.89	0.04	2.98	0.02	0.97	0.01
ATFT	3.10	0.29	2.34	0.23	0.88	0.07
CTFT	3.00	0.00	2.77	0.18	0.99	0.00
GTFT $\epsilon = 0.05$	2.94	0.04	2.92	0.07	0.97	0.02
GTFT $\epsilon = 0.1$	2.89	0.20	2.83	0.27	0.92	0.14
GTFT $\epsilon = 0.33$	2.89	0.11	2.41	0.55	0.70	0.33
HTF2T	3.77	0.06	1.61	0.03	0.97	0.00
HTFT2	2.88	0.04	2.97	0.02	0.97	0.01
HTFT3	2.84	0.07	3.00	0.03	0.96	0.01
NP	2.48	0.13	2.85	0.12	0.82	0.13
NPM	2.94	0.05	2.94	0.06	0.97	0.03
OTFT	1.89	1.04	2.29	0.65	0.96	0.02
RP $\epsilon = 0.01$	2.91	0.07	2.94	0.07	0.97	0.01
RP $\epsilon = 0.1$	2.44	0.12	2.63	0.19	0.67	0.18
RTFT	4.92	0.04	0.04	0.03	0.98	0.00
STFT	2.94	0.03	2.96	0.03	0.99	0.00
TF2T	3.88	0.04	1.56	0.02	0.97	0.00
TFT	2.96	0.01	2.96	0.01	0.98	0.00

weakness. Others require sequences of cooperations, and, yet again, Tages is able to produce them.

The player *Hard Tit for Tat* with a window of three is a paradigmatic example: Tages is able to predict its behavior 96% of the times, and select the best line of play, although the model is seldom exact (Fig. VI-B). On a closer examination, all ND-FSMs include three states where the opponent defects. Tages shows no interest in determining exactly how these three states are connected, because against all models the best option is to always cooperate, and this is correct.

Strategies derived from *Tit for Tat* that are also adaptive will be discussed in the following.

C. Adaptive opponents

Table V reports the results against common adaptive players (for details see Appendix A). While able to adapt their own strategy to the opponent's, these players are definitely rational. Thus, it should not be a surprise that the performance of the modeling unit is particularly good.

TABLE V
PERFORMANCE AGAINST WELL-KNOWN "ADAPTIVE" STRATEGIES: AVERAGE PAYOFFS PER TURN AND MODEL EFFICACY (AVERAGE AND STANDARD DEVIATION OVER 10 GAMES).

Strategy	Tages		Opponent		Model	
	$\overline{P}$	P_σ	$\overline{P}$	P_σ	$\overline{E}$	E_σ
ADP	0.97	0.03	1.49	0.12	0.98	0.00
APAV	1.89	0.78	2.41	0.50	0.79	0.20
ATFT	3.26	0.21	2.28	0.31	0.92	0.05
CTFT	3.00	0.00	2.83	0.13	0.99	0.00
FBF	3.00	0.00	2.83	0.11	0.99	0.00
GRD	2.82	0.14	2.96	0.06	0.93	0.04
HM	2.72	1.18	1.85	0.47	0.97	0.04
OTFT	1.85	1.07	2.24	0.70	0.96	0.02
PAV	4.97	0.02	0.04	0.02	0.99	0.00
SG	2.91	0.04	2.97	0.02	0.97	0.01
SM	3.59	0.20	1.80	0.16	0.95	0.03

Several strategies may turn to *Always Defect* upon certain conditions, and Tages models this fact through a sink state. *Omega Tit for Tat* is a paradigmatic example: depending on the beginning of the game, it may play an endless sequence of defections. Whether this is the case, the only viable option for Tages is to defect as well; however, other times, *OTFT* adopts a *TFT*-like strategy. The direction taken by the game is randomly determined in the first moves. The two typical models built by Tages are shown in Fig. 4. The standard deviation, remarkably higher than in other experiments, is coherent with the analysis.

Table VI reports the results against the *zero-determinant* strategies (for details see Appendix A). As expected, Tages performs poorly: the goal of such strategies is not to maximize their own score, but rather to impose a certain relationship between the payoffs, thus violating the implicit assumption that the opponent is maximizing its own interest. Interestingly, the model unit does not perform badly, showing that Tages did build an almost reliable model.

TABLE VI
PERFORMANCE AGAINST "ZERO-DETERMINANT" STRATEGIES: AVERAGE PAYOFFS PER TURN AND MODEL EFFICACY (AVERAGE AND STANDARD DEVIATION OVER 10 GAMES).

Strategy	Tages		Opponent		Model	
	$\overline{P}$	P_σ	$\overline{P}$	P_σ	$\overline{E}$	E_σ
EXT2	1.36	0.40	1.67	0.60	0.74	0.27
ZDE	1.18	0.22	1.64	0.70	0.78	0.28
ZDF	1.98	0.08	2.14	0.15	0.46	0.21
ZDG	2.78	0.15	2.57	0.32	0.82	0.15

Finally, to assess the performance against an highly adaptive opponent, Table VII reports the match of independent copies of Tages. Four different experiments were performed: one against an identical copy of the program, using exactly the same parameters specified in Table I. Then, three experiments against impaired copies: one allowed to evolve only for one tenth of the generations ($\mathcal{G} = 30$); one using one tenth of the individuals ($\mu = 10$); and one using one tenth of the individuals for one tenth of the generations ($\mathcal{G} = 30$ and $\mu = 10$).

TABLE VII
PERFORMANCE AGAINST TAGES VITH DIFFERENT PARAMETERS: AVERAGE PAYOFFS PER TURN AND MODEL EFFICACY (AVERAGE AND STANDARD DEVIATION OVER 10 GAMES).

Parameters		Tages		Opponent		Model	
$\mathcal{G}$	μ	$\overline{P}$	P_σ	$\overline{P}$	P_σ	$\overline{E}$	E_σ
300	100	1.41	0.12	1.37	0.11	0.23	0.08
30	100	1.30	0.14	1.32	0.12	0.08	0.06
300	10	1.49	0.15	1.38	0.15	0.12	0.05
30	10	1.62	0.11	2.06	0.13	0.32	0.09

The problems of co-evolution are well known both in zero- and nonzero-sum games [**?**], and Tages performance are not particularly good in this scenario. In the long run, it adopts an always-defect strategy, and, consequently, models the other self as *AC*. Thus, it may be concluded that Tages cannot be reliably modeled by itself.

Interestingly, however, the copy with the most impaired modeling unit is the one with the best performances. Thus,

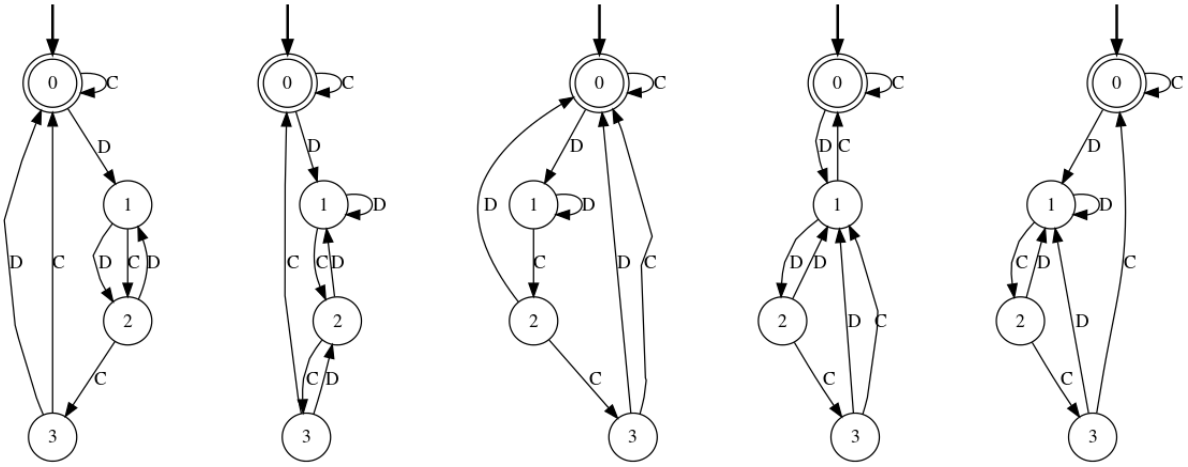

Fig. 3. Alternative models for *Hard Tit for Tat* ($w = 3$). Only the rightmost one is exact, nevertheless all models are functional: Tages correctly predicts its next move more than 95% of the times, and the brute-force analyzer suggests to always cooperate. See Fig. 2 for the definition of the symbols.

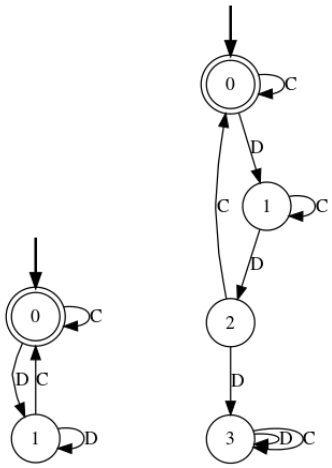

Fig. 4. Alternative models produced for the *Omega Tit for Tat*. Tages may recognize it as a simple *Tit for Tat* and cooperates. In some match, however, the opponent reaches the sink state 3 and the game cannot be recovered. See Fig. 2 for the definition of the symbols.

counter intuitively, the victory goes to whoever has the more simple model to represent an opponent. This can be explained, because impairing the model optimization results in far more simple ND-FSMs (three or four nodes, on average, against some tens of nodes). Such tiny models are, actually, *more* useful in the scenario: they are not *overfitted*, and the playing unit often resorts to its effective heuristic, generating a weighted random move (see Section V-C).

Remarkably, using $\mathcal{G} = 30$ and $\mu = 10$ would lower the overall performances. There is an evident relationship between the complexity of the opponent and the optimal complexity of the model. Tages tries to limit the complexity of its models according to the percentage of correct predictions using the internal threshold ζ, but this might not be effective enough when facing highly-complex strategies.

D. Round-Robin Tournaments

Following the methodology proposed in [**?**], the performances of Tages were assessed in round-robin tournaments: several random tournaments are played, each confronting exactly N different strategies randomly selected from the ones listed in table VIII. Similarly to Axelrod's, in the tournaments all players are matched against all others for five different games, and the total payoff collected during all games was considered as the final score. The player with the highest score is considered the winner of the tournaments.

The sizes of the tournaments ranged from five to thirty in steps of five. For each given size, one hundred thousand different tournaments were simulated. As other players, Tages was selected only in a fraction of the tournaments. Table VIII reports the percentages of tournaments won by the different players over the tournaments they participate into, and Table IX shows more details about the conduct in the single matches.

Tages is consistently able to win the majority of tournaments it takes part in. No other strategy has a similar performance, *Generous Tit for Tat* and *Omega Tit for Tat* are the only other players able to reach acceptable results in all tournament sizes, but their percentages of victories are a fraction of Tages'.

Some players, like *Omega Tit for Tat* and *Soft Majority*, show good performances in small tournaments only. Such strategies may be able to score more points than their opponents in one-to-one confrontations, but also reduce both payoffs. Thus, over several matches, they are likely to receive a smaller total payoff.

Following the idea of Axelrod's round-robin tournaments, Tages optimizes its global payoff and not the number of games it scores more than its opponent. Indeed, the mechanisms of *reconciliation* and *exploration* was added to this precise end: despite the fact that the model is highly effective, Tages may choose a sub-optimal move hoping to increase its global payoff in the long run.

Increasing the number of random strategies, for instance adding opponents from Table II or Table III, would further

TABLE VIII
PERFORMANCE IN ROUND-ROBIN TOURNAMENTS: PERCENTAGE OF VICTORIES IN THE DISPUTED TOURNAMENTS (AVERAGE OVER $100,000$ RANDOM TOURNAMENTS FOR EACH TOURNAMENT SIZE).

Player	Tournament size					
	5	10	15	20	25	30
2TFT	12.4	0.9	0.1	0.0	0.0	0.0
AC	28.1	10.4	2.8	0.5	0.0	0.0
AD	19.3	1.1	0.0	0.0	0.0	0.0
ADP	33.0	17.1	7.7	2.4	0.3	0.0
APAV	19.5	5.0	1.2	0.1	0.0	0.0
ATFT	29.4	10.3	3.5	0.7	0.1	0.0
CTFT	20.3	6.7	2.2	0.6	0.1	0.0
EXT2	1.6	0.0	0.0	0.0	0.0	0.0
FBF	19.7	6.7	2.2	0.6	0.1	0.0
FRT3	12.1	1.5	0.1	0.0	0.0	0.0
FRT4	15.1	2.2	0.2	0.0	0.0	0.0
GRD	26.5	14.3	6.8	2.4	0.5	0.0
GRM	19.5	7.7	1.6	0.1	0.0	0.0
GTFT $\epsilon = 0.05$	32.5	25.4	18.9	12.4	6.4	2.6
GTFT $\epsilon = 0.1$	36.3	33.1	29.4	23.1	16.7	10.5
GTFT $\epsilon = 0.33$	40.0	36.0	31.4	25.0	18.1	10.8
HM	2.0	0.0	0.0	0.0	0.0	0.0
HTF2T	26.4	9.2	2.4	0.4	0.0	0.0
HTFT2	13.2	1.3	0.1	0.0	0.0	0.0
HTFT3	17.9	3.1	0.5	0.0	0.0	0.0
NP	2.7	0.0	0.0	0.0	0.0	0.0
OTFT	36.2	30.3	27.3	22.9	15.6	6.9
PAV	29.4	11.5	3.6	0.7	0.1	0.0
PRO	1.1	0.0	0.0	0.0	0.0	0.0
RND	6.1	0.1	0.0	0.0	0.0	0.0
RP $\epsilon = 0.01$	6.5	0.2	0.0	0.0	0.0	0.0
RP $\epsilon = 0.1$	5.1	0.1	0.0	0.0	0.0	0.0
RTFT	11.0	1.2	0.0	0.0	0.0	0.0
SG	25.2	12.7	6.3	2.7	0.8	0.1
SM	36.5	20.9	11.8	5.6	1.8	0.3
STF2T	31.3	12.0	3.5	0.8	0.1	0.0
STFT	3.0	0.1	0.0	0.0	0.0	0.0
Tages	**75.5**	**75.7**	**78.7**	**82.8**	**87.2**	**92.4**
TFT	18.6	5.0	1.5	0.3	0.0	0.0
ZDE	2.1	0.0	0.0	0.0	0.0	0.0
ZDF	2.8	0.0	0.0	0.0	0.0	0.0
ZDG	22.4	8.9	2.9	0.6	0.1	0.0

boost the advantage of Tages, as it can be seen from the global results.

VII. CONCLUSIONS AND FUTURE WORKS

This paper presented Tages, an adaptive player for the IPD. Differently from similar strategies, it does not require a set of pre-determined archetypes to choose from, nor any distinction between *exploration* and *exploitation* phases. During the game Tages builds a model of its current adversary as a ND-FSM, and, at the same time, exploits it to push the game into the most favorable line of exchanges. Models are created from scratch and optimized by an EA to be *coherent*, *competitive* and *compact*. Such models need to be functional in predicting the opponent's replies, rather than to be exact or complete.

Experimental results demonstrated that Tages is consistently able to prevail in tournaments of different sizes against strong opponents taken from the literature. As Tages's goal is to maximize its own payoff, regardless of the score of the opponent, in one-to-one matches it overcame some two-thirds of the players. But its performances in round-robin competitions are thirty-five or more percentage points better than all the other strategies.

Experimental evidence also suggests that increasing the complexity and size of the models is not beneficial, as the ND-FSMs get more coherent with the past record of the game but less effective at predicting its future development. This phenomenon could be similar to the *overfitting* problem in linear regression. For this reason, Tages included a very tight mechanism to prevent bloating.

Similarly, decreasing the complexity and size of the models would reduce the effectiveness against all opponents that are correctly modeled. Indeed, changing significantly the internal parameters, either strengthening or impairing the modeling unit, is likely to decrease the performance in the tournaments. However, self-adapting mechanisms for better tackling highly-complex strategies could be added in future works.

Future works also include adapting Tages to evolutionary IPD competitions, as getting more payoff than the opponent in each match could be significantly different from getting the total highest payoff.

VIII. ACKNOWLEDGMENTS

Computational resources were provided by HPC@POLITO, a project of the Department of Control and Computer Engineering (DAUIN) at Politecnico di Torino[13].

APPENDIX

2TFT (*Two Tits For Tat*) — Cooperates unless the opponent defects. To retort, it defects twice. Then, if the opponent cooperated, it starts cooperating again.

AC (*Always Cooperate*) — Always cooperates.

AD (*Always Defect*) — Always defects.

ADP (*Adaptive*) — Starts with a sequence of five consecutive cooperations, followed by five defections; from the 11^{th} turn, it chooses the move that has been most profitable so far.

APAV (*Adaptive Pavlov*) — Categorizes the opponent's behavior using four classes, and then plays accordingly. If the opponent is *fully cooperative*, APAV behaves like a simple *TFT*; if it is *almost cooperative*, APAV plays *Soft Tit for Two Tats* in order to recover mutual cooperation; if *aggressive* or *random*, APAV always defects. To categorize the opponent, APAV plays six turns as *TFT*. If the opponent started with a cooperation, it is identified as *fully cooperative*; if the opponent defected three times, it is identified as *almost cooperative*; if the opponent defected four or more times, it is identified as *aggressive*; in all other cases, the opponent is considered *random*. In order to deal with the situations in which the opponents may change their actions, the average payoff is computed every six turns. If it is lower than $\mathbf{R}$, the process of opponent identification is restarted.

ATFT (*Adaptive Tit for Tat*) — Updates the variable w in $[0, 1]$ according to the opponent's moves: w is slowly pushed toward 1 on cooperations, toward 0 on defections. At each turn, if $w \geq 0.5$ *ATFT* cooperates; otherwise, it defects. The initial value are $w = 0.5$ and $r = 0.99$ for the adaption rate.

CCD (*Periodically CCD*) — Cooperates twice and then defects; then repeats the pattern.

[13]http://www.hpc.polito.it

CD (*Periodically CD*) — Cooperates and then defects; then repeats the pattern.

CS (*Collective Strategy*) — Starts cooperating and defects in the second turn. If the opponent played the same moves, *CS* starts behaving as *TFT*; otherwise, it plays AD.

CTFT (*Contrite Tit for Tat*) — Same as *TFT* when there is no noise. In a noisy environment, once it receives **T** because of an error, it will choose cooperate twice in order to recover mutual cooperation.

DDC (*Periodically DDC*) — Defects twice and then cooperates; then repeats the pattern.

EXT2 (*Zero-Determinant Extort-2*) — Let S_{ZD} be the total payoff earned by the ZD strategy, and S_o the one of its opponent, the strategy imposes the linear relationship $S_{ZD} - \mathbf{P} = 2 \cdot (S_o - \mathbf{P})$. That is, guarantees to EXT2 twice the share of payoffs above the "punishment" threshold, compared to those received by the current opponent.

FBF (*Firm But Fair*) — Cooperates on the first turn, and cooperates except after receiving a S payoff.

FRT3 (*Fortress3*) — Tries to recognize a kin member by playing the sequence *defect*, *defect*, *cooperate*. If the opponent plays the same sequence, it cooperates until the opponent defects. Otherwise, it defects until the opponent defects on continuous two moves, and then it cooperates on the following move.

FRT4 (*Fortress4*) — Tries to recognize the opponent by playing the sequence *defect*, *defect*, *defect*, *cooperate*. If the opponent plays the same sequence, it cooperates until the opponent defects. Otherwise, if it does not recognize the opponent as a friend, it defects until the opponent defects for three consecutive moves, then cooperates on the next.

FS (*Fair strategy*) — Defects with a probability p equals to the frequency of defections played by the opponent; cooperates with probability $1 - p$. In the first turns the probability is smoothed towards 0.5.

GRD (*Gradual*) — Cooperates on the first turn, and cooperates as long as the opponent cooperates; after the first defection of the other player, it defects one time and cooperates 2 times; ... after the *n*-th defection, it reacts with *n* consecutive defections and then plays two cooperate moves.

GRM (*Grim Trigger*) — cooperates until the opponent defects, and subsequently always defects. The strategy is also known as *Grudger* or *Spiteful*.

GTFT (*Generous Tit for Tat*) — Acts as a simple *TFT*, except that it cooperates with a probability q instead of just retaliating after the opponent defects. The parameter ϵ is usually small, typically $\epsilon = 0.1$. Some authors propose $\epsilon = min(1 - \frac{\mathbf{T}-\mathbf{R}}{\mathbf{R}-\mathbf{S}}, \frac{\mathbf{R}-\mathbf{P}}{\mathbf{T}-\mathbf{P}})$, i.e., $\epsilon = 0.33$ with the standard payoff matrix. The strategy is also known as *Naïve Peace Maker* or *Soft Joss*.

HM (*Hard Majority*) — Defects on the first turn, and defects if the number of defections of the opponent is greater than or equal to the number of times it has cooperated; otherwise it cooperates.

HS (*Handshake*) — Defects on the first turn and cooperates on the second. If the opponent behaves the same, it always cooperates. Otherwise, it always defects.

HTF2T (*Hard Tit For Two Tats*) — Cooperates unless the opponent plays two consecutive defections, then keeps defecting unless the adversary plays two consecutive cooperations. Then it starts cooperating again, and so on.

HTFT2 (*Hard Tit for Tat (2-turn window)* — Cooperates on the first turn, then defects only if the opponent has defected in any of the previous two turns. As some sources report the size of the window to be three turns instead of two, both versions have been included here.

HTFT3 (*Hard Tit for Tat (3-turn window)* — Cooperates on the first turn, then defects only if the opponent has defected in any of the previous three turns. See the discussion on the previous entry.

NP (*Naïve Prober*) — Acts as simple *TFT*, but defects unprovoked with a probability ϵ. The parameter ϵ is usually small, typically $\epsilon = 0.1$. The strategy is also known as *Hard Joss*.

OTFT (*Omega Tit for Tat*) — Normally behaves like *TFT*, but it is ready to play an extra cooperation for breaking deadlocks, i.e., two interlaced sequences of d alternating cooperation/defection. Moreover, *OTFT* measures the *randomness* of the opponent counting the number of times it changes behavior, and turns to an *Always Defect* if the value exceeds a given threshold t. Typically, $d = 3$ and $t = 8$.

PAV (*Pavlov*) — Cooperates on the first turn; if a payoff of R or T is received in the last turn then repeats last choice, otherwise chooses the opposite one.

PRO (*Prober*) — Starts with a sequence of one defection followed by two cooperations. If the opponent cooperated in the second and third turn, it keeps defecting for the rest of the game; otherwise, it plays as *TFT*.

RND (*Random Player*) — Randomly chooses between cooperation and defection with equal probability, with no memory of previous exchanges.

RP (*Remorseful Prober*) — Acts as simple *TFT*, but occasionally defects with a probability ϵ. Unlike *Naïve Prober*, however, it does not retort if the opponent defects after its unfair move. The parameter ϵ is usually small, typically $\epsilon = 0.1$.

RTFT (*Reverse Tit for Tat*) — Starts defecting, and then chooses the opposite of the opponent's previous action. This apparently illogical variant of *Tit for Tat* is also known as *Psycho*.

SGS (*Southampton Group strategies*) — A group of strategies are designed to recognize each other through a predetermined sequence of 5-10 moves at the start. Once two SGSs recognize each other, they will act as a *master* or *slave* - a master will always defect while a slave will always cooperate in order for the master to win the maximum points. If the opponent is not recognized as *SGS*, it will behave like an *AD* to minimize the score of the opponent.

SG (*Soft Grudger*) — Cooperates until the opponent defects. In this case, it punishes the behavior with a sequence of four defections. Then, it offers a peace agreement with two consecutive moves of cooperation.

SM (*Soft Majority*) — Cooperates on the first turn, and cooperates as long as the number of times the opponent has

cooperated is greater than or equal to the number of times it has defected; otherwise, it defects.

STF2T (*Soft Tit For Two Tats*) — Cooperates unless the opponent plays two consecutive defections.

STFT (*Suspicious Tit for Tat*) — starts defecting, and then replicates the opponent's previous action. The strategy is also known as *Evil Tit For Tat*.

TFT (*Tit for Tat*) — Cooperates on the first turn, copies the opponent's last move afterwards.

ZDE (*Zero-Determinant Extort*) — let S_{ZD} be the total payoff earned by the ZD strategy, and S_o the one of its opponent, the strategy imposes the linear relationship $S_{ZD} + \mathbf{P} = 3 \cdot (S_o - \mathbf{P})$.

ZDF (*Zero-Determinant Fixed Score*) — let S_{ZD} be the total payoff earned by the ZD strategy, and S_o the one of its opponent, the strategy tries to fix the opponent's score to a pre-determined value g. Usually, with the standard payoff matrix, $g = 2$.

ZDG (*Zero-Determinant Generous*) — let S_{ZD} be the total payoff earned by the ZD strategy, and S_o the one of its opponent, the strategy enforces the relationship $S_{ZD} = 2 \cdot (S_o - \mathbf{R})$ between the two strategies' scores. Compared to EXT2, it offers the opponent a higher portion of the payoffs.

REFERENCES

[1] E. N. Zalta and S. Abramsky, "Stanford encyclopedia of philosophy," 2003.

[2] W. Poundstone and N. Metropolis, "Prisoner's dilemma: John von neumann, game theory, and the puzzle of the bomb," *Physics Today*, vol. 45, p. 73, 1992.

[3] P. Pettit and R. Sugden, "The backward induction paradox," *The Journal of Philosophy*, pp. 169–182, 1989. [Online]. Available: http://www.jstor.org/stable/2026960

[4] M. A. Nowak and R. M. May, "Evolutionary games and spatial chaos," *Nature*, vol. 359, no. 6398, pp. 826–829, 1992.

[5] D. P. KRAINES and V. Y. KRAINES, "Natural selection of memory-one strategies for the iterated prisoner's dilemma," *Journal of Theoretical Biology*, vol. 203, no. 4, pp. 335–355, 2000.

[6] L.-A. Barlow and D. Ashlock, "The impact of connection topology and agent size on cooperation in the iterated prisoner's dilemma," in *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013, pp. 1–8.

[7] S. Bankes, "Exploring the foundations of artificial societies: Experiments in evolving solutions to iterated n-player prisoners dilemma," in *Artificial life IV*. Cambridge, MA: MIT Press, 1994, pp. 337–342.

[8] D. Ashlock, C. Kuusela, and M. Cojocaru, "Shopkeeper strategies in the iterated prisoner's dilemma," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 1063–1070.

[9] R. Axelrod, "Effective choice in the prisoner's dilemma," *Journal of Conflict Resolution*, vol. 24, no. 1, pp. 3–25, 1980.

[10] W. Ashlock, J. Tsang, and D. Ashlock, "The evolution of exploitation," in *Foundations of Computational Intelligence (FOCI), 2014 IEEE Symposium on*. IEEE, 2014, pp. 135–142.

[11] J. Li, P. Hingston, and G. Kendall, "Engineering design of strategies for winning iterated prisoner's dilemma competitions," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 4, pp. 348–360, 2011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6004823

[12] E. Piccolo and G. Squillero, "Adaptive opponent modelling for the iterated prisoner's dilemma," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 836–841.

[13] M. Gaudesi, E. Piccolo, G. Squillero, and A. Tonda, "Turan: Evolving non-deterministic players for the iterated prisoners dilemma," in *Evolutionary Computation (CEC), Proceedings of the 2014 IEEE Congress on*, 2014.

[14] R. M. Axelrod, *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton University Press, 1997.

[15] D. B. Fogel, "Evolving behaviors in the iterated prisoner's dilemma," *Evolutionary Computation*, vol. 1, no. 1, pp. 77–97, 1993.

[16] N. Franken and A. Engelbrecht, "Particle swarm optimization approaches to coevolve strategies for the iterated prisoner's dilemma," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 6, pp. 562–579, 2005.

[17] P. J. Darwen and X. Yao, "Co-evolution in iterated prisoner's dilemma with intermediate levels of cooperation: Application to missile defense," *International Journal of Computational Intelligence and Applications*, vol. 2, no. 01, pp. 83–107, 2002.

[18] G. Kendall, X. Yao, and S. Y. Chong, *The Iterated Prisoners' Dilemma: 20 Years on*. World Scientific Publishing Co., Inc., 2007.

[19] T. W. Sandholm and R. H. Crites, "Multiagent reinforcement learning in the iterated prisoner's dilemma," *Biosystems*, vol. 37, no. 1, pp. 147–166, 1996.

[20] A. Rubinstein, "Finite automata play the repeated prisoner's dilemma," *Journal of economic theory*, vol. 39, no. 1, pp. 83–96, 1986.

[21] D. Ashlock and E.-Y. Kim, "Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 5, pp. 647–659, 2008.

[22] J. Tsang, "The parametrized probabilistic finite-state transducer probe game player fingerprint model," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 2, no. 3, pp. 208–224, 2010.

[23] A. Rogers, R. K. Dash, S. D. Ramchurn, P. Vytelingum, and N. R. Jennings, "Coordinating team players within a noisy iterated prisoners dilemma tournament," *Theoretical Computer Science*, vol. 377, no. 1, pp. 243–259, 2007.

[24] W. Slany and W. Kienreich, "On some winning strategies for the iterated prisoner's dilemma or mr. nice guy and the cosa nostra," *arXiv preprint cs/0609017*, 2006.

[25] J. Li, "How to design a strategy to win an ipd tournament," *The Iterated Prisoners Dilemma*, vol. 20, pp. 89–104, 2007.

[26] P. Hingston, D. Dyer, L. Barone, T. French, and G. Kendall, "Opponent modelling, evolution, and the iterated prisoner's dilemma," *The Iterated Prisoner's Dilemma: Celebrating the 20th Anniversary, World Scientific*, pp. 139–170, 2007.

[27] W. H. Press and F. J. Dyson, "Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent," *Proceedings of the National Academy of Sciences*, vol. 109, no. 26, pp. 10 409–10 413, 2012.

[28] C. Adami and A. Hintze, "Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything," *Nature Communications*, vol. 4, p. Online, August 2013.

[29] A. J. Stewart and J. B. Plotkin, "From extortion to generosity, evolution in the iterated prisoner's dilemma," *Proceedings of the National Academy of Sciences*, 2013.

[30] M. O. Rabin and D. Scott, "Finite automata and their decision problems," *IBM journal of research and development*, vol. 3, no. 2, pp. 114–125, 1959.

[31] J. C. Martin, *Introduction to Languages and the Theory of Computation*. McGraw-Hill New York, 2003, vol. 2.

[32] G. Greewood, G. B. Fogel, and M. Ciobanu, "Emphasizing extinction in evolutionary programming," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1. IEEE, 1999.

[33] J. Tsang, "Fingerprint multilateration for automatically classifying evolved prisoner's dilemma agents," in *Foundations of Computational Intelligence (FOCI), 2014 IEEE Symposium on*. IEEE, 2014, pp. 98–105.

[34] A. W. F. Edwards, *Likelihood*. CUP Archive, 1984.

[35] G. Squillero, "Chromatic selection - an oversimplified approach to multi-objective optimization," in *Applications of Evolutionary Computation. Proceedings of EvoAPPS.*, 2015, pp. 640–649.

[36] C. H. Pence and L. Buchak, "Oyun: A new, free program for iterated prisoner's dilemma tournaments in the classroom," *Evolution: Education and Outreach*, vol. 5, no. 3, pp. 467–476, 2012.

[37] D. Ashlock, W. Ashlock, S. Samothrakis, S. Lucas, and C. Lee, "From competition to cooperation: Co-evolution in a rewards continuum," in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*. IEEE, 2012, pp. 33–40.

TABLE IX
PERFORMANCE IN ONE-TO-ONE MATCH: PAYOFFS PER GAME AND MODEL EFFICACY (AVERAGE AND STANDARD DEVIATION OVER 10 MATCHES, 5 GAMES EACH).

Opponent name	Game 1				Game 2				Game 3				Game 4				Game 5				Model	
	$\overline{P}^T$	P^T_σ	$\overline{P}^O$	P^O_σ	$\overline{P}^T$	P^T_σ	$\overline{P}^O$	P^O_σ	$\overline{P}^T$	P^T_σ	$\overline{P}^O$	P^O_σ	$\overline{P}^T$	P^T_σ	$\overline{P}^O$	P^O_σ	$\overline{P}^T$	P^T_σ	$\overline{P}^O$	P^O_σ	$\overline{E}$	E_σ
2TFT	484	5.4	498	5.3	1060	8.2	1080	2.2	906	7.7	921	1.8	1835	10.2	1874	2.5	945	5.4	960	1.3	1.00	0.001
ADP	164	4.2	259	18.6	378	25.5	411	21.0	319	5.2	331	19.5	629	4.5	666	18.6	328	18.7	356	71.2	0.77	0.289
APAV	377	128.6	383	74.8	658	357.2	739	198.6	370	90.5	633	198.9	673	100.3	1195	341.2	333	54.5	579	158.1	0.29	0.275
ATFT	518	30.3	425	66.5	1196	140.8	711	57.5	853	331.4	620	83.9	1721	691.7	1260	152.1	854	410.9	625	73.6	0.62	0.299
CTFT	504	0.0	483	12.0	1077	0.0	1056	10.2	918	0.3	898	9.1	1866	0.6	1835	11.6	957	0.3	943	9.7	1.00	0.000
EXT2	311	23.3	431	32.2	533	57.8	723	117.5	413	47.2	521	71.7	808	55.5	984	130.9	389	36.8	477	57.3	0.05	0.023
FBF	504	0.3	470	28.0	1077	0.0	1056	9.7	918	0.0	907	11.6	1866	0.6	1834	11.9	957	0.3	941	9.2	1.00	0.000
FRT3	484	11.2	493	10.8	1059	11.9	1056	11.4	906	7.3	894	9.1	1850	5.4	1825	16.6	941	8.9	932	11.1	1.00	0.000
FRT4	450	23.9	492	6.6	1046	12.4	1034	18.6	892	16.0	883	14.3	1827	23.2	1806	17.6	940	6.7	929	12.5	1.00	0.001
GRD	472	31.2	494	6.6	1017	46.2	1068	12.8	888	35.7	911	10.2	1734	116.7	1893	66.1	898	85.2	963	52.0	0.93	0.079
GRM	150	5.3	259	20.1	491	165.4	531	153.7	393	114.7	412	105.5	759	105.0	802	102.1	474	153.3	481	149.0	1.00	0.001
GTFT $\epsilon = 0.05$	494	5.7	494	7.1	1068	10.7	1066	14.1	913	4.9	912	4.7	1857	9.4	1854	14.3	942	27.6	935	34.5	0.99	0.010
GTFT $\epsilon = 0.1$	496	4.3	491	7.9	1057	29.3	1043	44.1	883	48.4	863	73.6	1800	78.1	1763	113.0	920	48.8	895	79.7	0.83	0.258
GTFT $\epsilon = 0.33$	491	8.5	436	88.7	1023	76.3	779	218.6	871	55.8	633	198.5	1728	138.3	1227	413.1	885	70.3	586	211.7	0.22	0.197
HM	496	158.8	383	103.6	986	509.7	646	206.7	785	418.6	498	111.0	1649	867.3	848	123.3	941	456.2	434	71.9	0.92	0.142
HTF2T	641	14.5	268	5.5	1416	14.0	550	6.5	1216	7.3	465	4.1	2460	14.6	953	9.9	1263	6.7	488	4.3	1.00	0.001
HTFT2	485	9.6	499	3.3	1065	6.5	1079	1.3	905	7.0	921	1.2	1847	8.4	1871	2.1	943	7.4	960	2.0	1.00	0.001
HTFT3	472	12.1	505	3.5	1046	20.6	1086	6.6	891	10.3	929	4.0	1822	15.1	1884	6.1	940	10.8	964	4.4	0.99	0.001
NP	416	24.8	477	18.4	844	64.9	955	75.7	680	67.3	765	81.8	1348	203.4	1529	271.8	683	126.9	779	160.4	0.28	0.226
OTFT	372	170.3	424	98.0	819	354.1	833	334.0	696	303.6	706	289.8	1460	569.7	1465	556.7	728	312.7	733	305.2	0.99	0.010
PAV	836	2.6	7	3.9	1787	3.4	13	5.1	1523	3.4	10	5.1	3098	4.0	18	6.1	1589	5.2	9	7.9	1.00	0.000
PRO	487	3.7	490	4.3	1070	1.8	1074	3.2	912	1.6	916	2.4	1856	4.8	1859	6.3	951	2.7	954	1.9	0.99	0.001
$\mathcal{R}_{0.0}$ (AD)	146	4.8	256	19.2	350	4.1	394	16.2	300	5.1	330	20.2	614	4.9	655	19.8	315	1.8	334	7.1	1.00	0.000
$\mathcal{R}_{0.1}$	186	20.0	331	70.6	467	10.1	411	49.1	420	7.2	315	25.2	889	61.9	606	19.0	447	38.6	304	26.7	0.04	0.030
$\mathcal{R}_{0.2}$	256	10.0	302	40.5	602	38.1	381	56.5	568	5.0	291	28.9	1083	58.0	578	56.5	581	50.7	284	14.0	0.02	0.014
$\mathcal{R}_{0.3}$	282	44.1	323	108.9	723	41.0	415	99.7	626	40.7	309	67.1	1321	68.1	571	74.3	683	20.6	269	27.1	0.02	0.016
$\mathcal{R}_{0.4}$	372	44.5	307	107.3	854	46.9	417	75.0	772	54.7	300	64.6	1534	11.0	564	59.2	777	23.2	276	51.9	0.02	0.011
$\mathcal{R}_{0.5}$ (RND)	427	40.0	267	47.8	993	53.8	378	87.6	865	36.0	274	35.3	1768	60.7	507	56.0	895	48.1	250	34.5	0.01	0.006
$\mathcal{R}_{0.6}$	482	40.4	267	53.6	1122	55.7	337	63.5	992	35.3	233	39.9	2040	31.7	436	30.4	1044	11.4	222	26.7	0.02	0.011
$\mathcal{R}_{0.7}$	622	9.6	127	29.7	1287	73.7	220	74.5	1123	10.4	165	26.6	2272	21.2	288	50.7	1153	13.4	153	24.5	0.02	0.005
$\mathcal{R}_{0.8}$	664	13.7	116	15.6	1468	29.3	157	31.9	1246	51.2	125	29.0	2550	26.6	240	22.0	1303	47.4	121	12.5	0.03	0.009
$\mathcal{R}_{0.9}$	732	29.7	83	38.8	1588	28.6	103	30.4	1362	14.8	89	16.5	2815	32.6	133	43.3	1439	8.0	57	11.1	0.04	0.014
$\mathcal{R}_{1.0}$ (AC)	835	1.8	8	2.7	1789	3.9	8	5.8	1523	5.2	11	7.7	3097	2.8	19	4.3	1587	4.2	12	6.4	1.00	0.000
RP $\epsilon = 0.01$	490	6.1	492	3.5	1049	18.9	1068	16.6	901	8.2	913	11.1	1823	22.0	1847	22.3	938	11.8	956	8.6	0.98	0.019
RP $\epsilon = 0.1$	407	14.9	433	22.6	810	63.6	820	101.4	638	64.8	610	97.0	1213	110.8	1131	174.8	601	54.9	541	79.3	0.10	0.045
SG	474	12.6	472	19.4	1058	12.3	1033	27.9	901	10.8	876	22.0	1843	10.3	1793	30.1	941	9.0	903	30.0	0.99	0.002
SM	530	187.2	285	37.5	1256	337.9	584	69.0	961	441.9	479	50.1	2032	790.5	925	104.6	1005	407.6	462	55.5	0.89	0.217
TFT/RTFT $\alpha = 0.0$ (TFT)	495	3.3	495	3.3	1074	1.6	1074	1.6	916	1.3	916	1.3	1859	2.5	1859	2.5	953	1.9	953	1.9	1.00	0.000
TFT/RTFT $\alpha = 0.1$	419	14.1	452	10.6	838	68.6	868	195.9	658	84.9	618	155.6	1225	170.4	1207	315.4	638	84.1	573	140.7	0.10	0.030
TFT/RTFT $\alpha = 0.2$	381	60.1	313	28.3	761	26.2	511	54.4	609	2.1	396	29.7	1228	4.9	738	181.7	608	49.5	358	35.4	0.01	0.000
TFT/RTFT $\alpha = 0.3$	390	28.6	337	43.4	825	22.5	510	90.8	689	20.4	349	33.4	1354	37.4	674	81.7	712	15.5	312	13.7	0.03	0.013
TFT/RTFT $\alpha = 0.4$	381	35.8	298	64.6	927	45.2	387	68.6	804	47.1	280	36.1	1579	68.5	539	66.6	812	22.1	275	14.7	0.02	0.011
TFT/RTFT $\alpha = 0.5$	438	20.0	238	79.6	1011	36.3	326	118.5	889	56.5	245	72.1	1799	80.4	463	114.9	952	12.2	219	45.2	0.01	0.007
TFT/RTFT $\alpha = 0.6$	500	39.6	199	53.3	1097	28.2	250	16.5	979	31.1	196	20.1	2066	64.8	346	43.6	1025	56.1	195	16.0	0.01	0.004
TFT/RTFT $\alpha = 0.7$	577	50.2	141	64.1	1297	11.1	190	46.1	1120	22.9	138	27.5	2257	26.6	296	61.5	1166	36.7	128	14.5	0.01	0.004
TFT/RTFT $\alpha = 0.8$	568	116.6	180	104.4	1378	154.2	201	113.0	1191	67.6	164	66.5	2468	110.6	271	122.0	1299	46.7	112	41.1	0.01	0.009
TFT/RTFT $\alpha = 0.9$	737	12.9	62	26.4	1596	60.2	96	43.5	1370	44.4	50	2.8	2774	84.8	114	52.2	1495	0.0	30	3.3	0.09	0.000
TFT/RTFT $\alpha = 1.0$ (RTFT)	822	6.1	12	4.4	1765	13.2	18	8.8	1505	13.8	15	9.2	3075	15.4	22	10.2	1574	6.3	13	4.2	1.00	0.000
STF2T	652	5.7	263	4.9	1430	2.2	545	3.8	1218	3.6	465	4.5	2473	3.9	946	5.6	1270	3.1	484	3.6	1.00	0.000
STFT	494	4.9	499	4.9	1071	1.6	1076	1.6	911	1.6	916	1.6	1856	2.7	1861	2.7	951	1.4	956	1.4	1.00	0.000
ZDE	263	45.2	421	36.3	482	49.1	650	97.6	369	44.1	480	86.7	724	51.5	897	97.8	351	40.6	428	57.6	0.06	0.054
ZDF	334	16.4	335	48.3	716	23.6	503	64.0	618	20.4	366	36.0	1233	43.3	703	72.0	629	17.8	327	29.8	0.03	0.008
ZDG	473	32.5	456	57.8	977	94.7	881	189.5	830	76.7	733	181.4	1631	201.4	1399	393.5	831	108.2	706	210.5	0.24	0.229